Andrea Hollung Nornes, Martine Alvilde Gran

Automatic Classification of Pro-Eating Disorder Twitter Accounts with Personality as a Feature

Master's thesis in Informatics
Supervisor: Björn Gambäck
May 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Kunnskap for en bedre verden

Andrea Hollung Nornes, Martine Alvilde Gran

Automatic Classification of Pro-Eating Disorder Twitter Accounts with Personality as a Feature

Master's thesis in Informatics
Supervisor: Björn Gambäck
May 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

A person or a group of people who considers eating disorders as a lifestyle, instead of a deadly mental disease, is called pro-eating disorder (abbreviated pro-ED). Eating disorders are the number one most deadly group of mental disorders and since the introduction of the internet, large online pro-ED communities have sprung forth. These communities share content focusing on eating disorder maintenance, inspiration, and motivation. Viewing this kind of content has been proven to be damaging, resulting in lower self-esteem and the desire to eat less. Multiple microblogging services such as Tumblr, Instagram and Pinterest have taken measures to limit the amount of pro-ED content. Twitter has not taken any measures, as of writing this thesis, which means that a lot of pro-ED related content is available on the site.

The goal of this thesis was to improve automatic classification of pro-ED Twitter accounts, by using the Big 5 personality model to calculate personality traits and add them to the list of features. A total of four datasets were accumulated, where two of the datasets ended up being used to train a Big 5 personality detection model and one was used to train a pro-ED classification model. The last dataset was found to significantly reduce the performance of the personality detection model, and was therefore discarded. The two datasets used to train the personality detection model were combined together and contained 2 636 Twitter accounts and essays. 169 of these were Twitter accounts, and the remaining 2 467 were essays. These accounts and essays were all labeled with Big 5 personality trait scores. The dataset used for the pro-ED classification model contained 6 824 Twitter accounts which were annotated as either *pro-ED*, *pro-recovery*, or *unrelated*.

After testing a number of features and machine learning algorithms, a new, state-of-the-art pro-ED classification model was created. This model takes the predictions from the personality detection model as a feature, in combination with unigrams, bigrams, and topic models. The algorithm used for creating the personality detection model was the Support Vector Regression algorithm and Global Vectors was used as the only feature. Both Support Vector Machine and Multilayer Perceptron were tested as the pro-ED classification algorithm. The best F_1 score was 0.99 and was achieved with the Multilayer Perceptron algorithm with the personality feature included in the feature set.

Sammendrag

En person eller en gruppe som anser spiseforstyrrelser som en livsstil, i stedet for en dødelig psykisk lidelse, omtales som å være pro-spiseforstyrrelse eller pro-ED (fra det engelske ordet pro-Eating Disorder). Spiseforstyrrelser er den gruppen psykiske lidelser med høyest dødsrate. Store pro-ED samfunn har vokst frem siden lanseringen av internett. Disse nettbaserte samfunnene deler innhold med fokus på opprettholdelse av spiseforstyrrelser samt deling av inspirasjon og motivasjon. Det har blitt bevist at å se på denne typen innhold fører til lavere selvtillit og et ønske om å spise mindre. Mange mikrobloggingtjenester, deriblant Tumblr, Instagram og Pinterest, har tatt grep for å redusere mengden pro-ED innhold. Twitter har, på det tidspunktet denne oppgaven ble skrevet, derimot ikke tatt grep for å fjerne slikt innhold, hvilket betyr at mye pro-ED innhold er tilgjengelig på denne plattformen.

Målet med denne oppgaven var å forbedre automatisk klassifisering av pro-ED kontoer på Twitter ved å ta i bruk personlighetstrekk fra Big 5 modellen som en feature. Totalt fire datasett ble samlet inn, der to ble brukt til å trene en Big 5 personlighets-detekteringsmodell, og ett ble brukt til å trene en pro-ED klassifiseringsmodell. Det siste datasettet ble ekskludert da det viste seg å påvirke resultatene på en negativ måte. De to datasettene som ble brukt til å trene personlighets-detekteringsmodellen ble slått sammen til ett stort dataset som inneholdt 169 Twitter kontoer og 2 467 essays. Disse kontoene og essayene hadde alle blitt merket med verdier for å representere Big 5 personlighetstrekk. Datasettet som ble brukt til klassifiseringen av pro-ED kontoer inneholdt 6 824 Twitter kontoer som ble merket med enten *pro-ED*, *pro-recovery*, eller *unrelated*.

Etter å ha testet en rekke features og maskinlæringsalgoritmer ble det laget en ny state-of-the-art modell for klassifisering av pro-ED kontoer på Twitter. Denne modellen tar resultatene fra personlighets-detekteringsmodellen som en feature, sammen med unigrams, bigrams og topic models. Algoritmen som ble brukt for personlighetsdetektering var Support Vector Regression med Global Vectors som feature. Både Support Vector Machine og Multilayer Perceptron ble testet som mulige algoritmer for pro-ED-klassifiseringsmodell av Twitter-kontoer. Den beste F_1 verdien var 0.99 og ble funnet ved å bruke Multilayer Perceptron med Big 5 personlighet inkludert i feature-settet.

Preface

This Master's Thesis was written during the Fall of 2018 and Spring of 2019 as a part of our Master of Science (MSc) degree in Informatics at the Department of Computer Science (IDI) at the Norwegian University of Science and Technology (NTNU).

We would like to thank our awesome supervisor, Björn Gambäck, for his help throughout the year. We would also like to say a huge thank you to the kind people at Bouvet Oslo, for providing us with a place to work, and to Kristin Hollung, for dedicating some of her time to give feedback on our thesis. Finally we would like to thank Ingrid Giæver for creating a new and interesting field of research and for letting us continue in her footsteps.

Andrea Hollung Nornes, Martine Alvilde Gran
Trondheim, 28th May 2019

Contents

List of Figures

List of Tables

Glossary

anorexia A restrictive eating disorder. Sufferers restrict the food intake. 1, 7, 8, 38

bulimia A compulsive eating disorder. Sufferers often binge eat then purge the body by throwing up. 1, 7, 8

microblogging service A web service providing online communication by short text snippets. 1, 2, 7, 10

thinspiration Images, music, and texts promoting the ideal thin body, also abbreviated thinspo. 1, 8, 37

vlog A video log where a person usually talks into a camera about daily life or interesting topics, similar to a diary. 41

wannarexic People who are not anorexic, but use pro-ED websites for inspiration to get the disease. 8, 38

Acronyms

BoW Bag of Words. 24, 25, 35, 39, 66, 67, 79, 107

CV Cross-Validation. 14

ED Eating Disorder. 1, 7–9

EDNOS Eating Disorder Not Otherwise Specified. 7

GIF Graphic Interchange Format. 10

GloVe Global Vectors. i, ii, 31, 36, 39, 57, 61, 64–69, 77–79, 83, 88, 89, 91, 95, 105

GP Gaussian Process. vi, 13, 15–17, 36, 58, 59, 64–70, 80–82, 87–89, 91, 105, 108

IDF Inverse Document Frequency. 26, 27

k-NN K-Nearest Neighbors. v, vi, ix, 15, 17, 18, 58, 59, 69, 80–82, 91

LDA Latent Dirichlet Allocation. 27, 39, 57

LIWC Linguistic Inquiry and Word Count. 30, 34–37, 58, 64, 65, 68, 74, 77, 79, 80, 88, 90, 106

LR Logistic Regression. 35, 37

MLP Multilayer Perceptron. i, ii, v, 15, 19, 58, 60, 69, 70, 80, 82–85, 91, 92, 96, 109

NLP Natural Language Processing. 15, 18, 23, 24, 31, 32, 53, 74

NLTK Natural Language Toolkit. 31

POS Part of Speech. 25, 26, 30–32, 34, 35, 37, 53, 57, 58, 74, 77, 79, 90, 108

QP Quadratic Programming. 15

RBF Radial Basis Function. 14, 68, 81

RegEx Regular Expression. 72, 74

RF Random Forest. 35, 37

RR Ridge Regression. v, 18, 58, 59, 69, 70, 80, 82, 91

SMO Sequential Minimal Optimization. 15, 35, 58

SVM Support Vector Machine. i, ii, vi, 13–16, 35–37, 39, 40, 58, 59, 61, 64, 65, 69, 71, 74, 75, 77, 80–84, 91, 108, 109

SVR Support Vector Regression. i, ii, 64–70, 83, 84, 87, 88, 95, 105, 106

TF Term Frequency. 26

TF-IDF Term Frequency-Inverse Document Frequency. 26, 56, 66, 67, 79, 107

1. Introduction

Pro-eating disorder (pro-ED) is a term which refers to a person with a positive view on eating disorders. Today it is possible to find communities of people identifying as pro-ED on a number of microblogging services such as Tumblr, Pinterest, Instagram, and Twitter. Members of the pro-ED communities use these platforms to share things like images, emotions and weight progress, as well as to give and receive support and motivation. This thesis focuses on the pro-ED accounts that can be found on Twitter and the Big 5 personality traits of these account owners. The information extracted from tweets and profile pages is analyzed and three different classifiers are built. The first classifier attempts to detect the personality of the Twitter account owner using the Big 5 personality model, the second attempts to detect pro-ED accounts on Twitter, while the third takes the predictions from the first model as a feature and uses it on the second model. This introductory chapter briefly presents the motivation behind this thesis (a further elaboration will be given in chapter 5). It also explains the research goal as well as giving a broader explanation of the goal in the form of three research questions. After this, the research method is explained along with the contributions this thesis provides. The final part of this chapter describes the structure of the rest of the thesis.

1.1. Motivation

Eating disorders (EDs) such as anorexia and bulimia are mental diseases which are difficult to cure. Studies have found that 4.6 % of the general Norwegian population have subclinical or clinical EDs, while in elite athletes the number rises to 13.5% (Sundgot-Borgen and Torstveit, 2004). Many people also go undiagnosed, suggesting the number might be even higher. In addition to affecting a broad portion of the world's population, eating disorders also have a high mortality rate, with anorexia nervosa having the highest mortality rate of all mental illnesses (Birmingham et al., 2005). Eating disorders are stigmatized illnesses and the sufferers often try to hide their disordered eating behaviors from their families and other social contexts (Yeshua-Katz and Martins, 2013).

The emergence of the internet and the ease of communication have given people with eating disorders a place to reach out and form communities outside of their normal social situations. While some of these communities focus on recovery, many tend to focus on maintaining the eating disorder, often in the form of sharing thinspiration, weight goals and progress. Viewing such pro-ED content can have a negative effect on the viewer in the form of lower social self-esteem, higher need to exercise and wrongly perceived weight (Bardone-Cone and Cass, 2007).

Microblogging services have been a popular place for posting pro-ED content and in 2012 both Tumblr, Pinterest, and Instagram changed their terms of service to prohibit content promoting self-harm. This included content that glorifies or promotes eating disorders (Tumblr, 2012; Pinterest, 2012). By prohibiting pro-ED content one expected the amount of this type of content to drop. However, the prohibition might not have had the anticipated effect. The pro-ED communities responded to the prohibition by becoming more secluded and inward-oriented, making them difficult to detect both for moderators, health services and family (Casilli, 2013).

One way to detect pro-ED content on a website is to have human moderators manually survey the page content. This takes time and proves to be ineffective considering the increasing amount of data being produced (Chancellor et al., 2017). Another method that has gained focus in the last few years is the automatic detection of web content. In automatic detection, a computer is trained to detect a certain type of content, often based on text analysis. This, however, is not straightforward. People are different and the nature of the content they produce vary.

The Big 5 personality model has been used to categorize personality for many years, including in research on people with eating disorders. When looking at the personality traits of eating disorder sufferers, researchers found that people with eating disorders have statistically significantly higher scores on the personality trait neuroticism, compared to control groups consisting of healthy people (Bollen and Wojciechowski, 2004). This means that personality could be a factor in differentiating between healthy people and people with eating disorders.

1.2. Goal and Research Questions

Based on the motivation described above the goal for this thesis is:

Goal *To improve upon automatic detection of pro-ED Twitter accounts by considering personality as a feature.*

Explained in more detail, the goal is to see if it is possible to improve upon the performance of automatic detection machine learning algorithms by using personality as a feature in addition to linguistic and non-linguistic features. In order to reach this goal, this thesis has been divided into three sub-goals, with the purpose of guiding the research in a structured manner towards the main goal. These sub-goals have been formulated as research questions and can be seen below.

Research question 1 (Personality) *Which machine learning model has the best potential for personality detection?*

The focus for this research question is to use related research to find promising machine learning models that can be used in the Big 5 personality categorization of Twitter account owners. The models will then be compared through experiments to see which models deliver the best results when it comes to performance. The performance is measured through the use of the Pearson correlation coefficient. In the experiments, different linguistic features will be tested along with the models in order to create the machine learning model with the best potential for personality detection.

Research question 2 (Pro-ED) *Which machine learning model has the best potential for pro-eating disorder classification?*

This research question is similar to research question 1, except that it focuses on finding the machine learning model with the best potential for pro-eating disorder classification. A baseline result will be created from an already existing pro-eating disorder model proposed by Giæver (2018). All other models found through related research will be compared to the baseline result by running performance experiments, as in research question 1. The performance in this case is measured through precision, recall and F_1 score values. The most promising machine learning model will be chosen as the model to be used further in the thesis.

Research question 3 *What impact does the inclusion of personality detection, as a feature, have on the performance of the pro-ED classifier?*

In order to be able to achieve the research goal, the two classification models from research questions 1 (personality) and 2 (pro-ED) will have to be combined into one functioning classification model. This research question aims to see how the performance of the pro-ED classification model is affected when introducing the results of the personality detection model as a feature.

1.3. Research Method

Research question 1 and 2 were answered by first conducting a study of related research and previous work relevant to the research field. For research question 1, related research and previous work related to personality detection were studied, while for research question 2, research and work related to pro-eating disorder in general, and on the Twitter platform, were studied. The thesis written by Giæver (2018) also lay the ground for the creation of a result baseline used in the research experiments. The literature review provided enough information to be able to run experiments which would answer the two research questions.

The third and last research question was answered by running a series of experiments and analyzing the results they produced. The experiments first looked at which features and classifiers would produce the best results for personality, using a dataset annotated with scores for the Big 5 personality traits. The same was done for pro-ED, with a dataset classified as either pro-ED or not. All the features in the experiments were tested one at a time, and the same was done for the classifiers. Finally, the best performing features and classifier were used to build a final pro-ED classifier that was tested with the feature set with and without the personality feature created by the personality detection model.

1.4. Contributions

Limited amount of research exists on the detection of pro-eating disorder in Twitter accounts. The work presented in this thesis contributes to the new field of research introduced by Giæver (2018). The experimental results show that classification of pro-ED accounts on Twitter can be done with an $F1$ score of 0.99, meaning that it can be used in the process of discovering pro-ED accounts automatically. Hopefully, this can be used to help reach out to people in need of help in a quick and effective way. The detection of pro-ED accounts is also an important stepping stone into removing content online that might cause harm to people. The experiments focusing on personality detection contribute to the expansion of the personality detection research field by using existing state-of-the-art personality detection methods in a new and innovative way.

1.5. Thesis Structure

The rest of the thesis has the following structure:

Chapter 2 - Eating Disorders and Personality: contains the background theory relevant to eating disorders and personality which is necessary to know in order to understand the terms and concepts mentioned throughout the thesis.

Chapter 3 - Machine Learning for Text Classification: provides information about machine learning concepts, the machine learning algorithms used for text classification in this thesis as well as classification metrics.

Chapter 4 - Text Representation, Annotation and Tools: explains the text representation models used as well as the concepts related to data annotation. The tools used as part of the thesis are also presented.

Chapter 5 - Related Work: discusses existing research related to the detection of personality in Twitter accounts, research on personality in eating disorder patients, as well as on automatic detection of pro-ED Twitter accounts.

Chapter 6 - Data: contains information about the four datasets used and how the data were processed.

Chapter 7 - Architecture: explains the feature extraction process and the architecture of the models created.

Chapter 8 - Experiments and Results: describes the experiment plan, setup and results.

Chapter 9 - Discussion and Evaluation: contains the evaluation of the experiments and discusses the research process and results as well as the limitations affecting the results. Ethical aspects surrounding this kind of research are also elaborated on.

Chapter 10 - Conclusion and Future Work: ends the thesis by posing a conclusion to the research. The conclusion takes the form of a summary of the answers to the research questions, and the goal result. In the end, the chapter proposes possibilities for future work.

2. Eating Disorders and Personality

This and the following two chapters present the information and theories needed in order to understand the content of this thesis. The focus for this chapter is to give an elaboration of pro-eating disorder as a term, as well as to explain related elements such as eating disorder, pro-eating disorder communities, and pro-eating disorder content. Pro-recovery is also mentioned as a counterpart. The Big 5 personality model, used in personality detection, is also explained, followed by a quick walk-through of the microblogging service, Twitter.

2.1. Pro-Eating Disorder

The term pro-eating disorder, often abbreviated pro-ED, references a movement that promotes a non-recovery oriented approach to eating disorders. Followers of this movement tend to describe the eating disorder as a lifestyle choice rather than a disease that needs to be treated (Fox et al., 2005). In order to fully understand the pro-ED concept, it is necessary to elaborate on the different elements that the pro-ED concept comprise of. These elements are eating disorders, the online pro-ED communities, pro-ED content, and pro-recovery communities.

2.1.1. Eating Disorders

Eating disorders (EDs) are fairly common diseases affecting as much as 4.6% of the general Norwegian population (Sundgot-Borgen and Torstveit, 2004). EDs are normally divided into four subgroups; anorexia nervosa, bulimia nervosa, binge eating disease and eating disorder not otherwise specified (EDNOS). These are different expressions of the disease, but common to all is the use of food as a means to handle emotional challenges. Whilst EDs are considered mental illnesses, they also have a large impact on the physical health of a person. The impact can be so great that it leads to long term or permanent health damages. EDs are serious diseases and have the highest mortality in all mental disorders (Birmingham et al., 2005). A variety of health issues such as osteoporosis, infertility, heart disease, brain disease and more are also possible outcomes for the people suffering from EDs.[1]

[1] `https://www.nationaleatingdisorders.org/`

A wide range of risk-factors can contribute to a person developing an ED. These involve biological risk factors such as diet history and family members with mental diseases, psychological factors like personality traits and body image, and sociological factors like bullying and weight-stigma in the media and culture. EDs often co-occur with other illnesses such as depression, anxiety, and substance abuse.

2.1.2. Pro-ED Communities

Online pro-ED communities exist on a number of forums and privately owned blogs as well as on social networking sites such as Twitter, Tumblr, Facebook, and Instagram. While communities of individuals with EDs have existed for many decades, through mailing letters and other forms of communication, the development of communication technologies, such as the internet, has nurtured the existence of pro-ED communities (Fox et al., 2005). As a result, the communities have flourished in recent years.

People with eating disorders join these communities for a sense of support and belonging and to talk to others who understand what they are going through. In the online communities they share content, related to their eating disorders, that is often directed towards how to keep up their eating disordered lifestyles. The communities can be further divided into sub-communities such as pro-anorexia (pro-ANA) and pro-bulimia (pro-MIA) which promote anorexia and bulimia respectively. While these are separated in some communities, they will all be counted as pro-ED in this thesis.

Many pro-ED communities give their members a feeling of having a collective identity (Whitehead, 2010), which they can be highly protective of. *Wannarexic* is a word often used to describe people who are not actually anorexic, but are on the pro-ED sites in order to get inspiration on how to get the disease. Being called a wannarexic is seen as an insult and members of the pro-ED communities often act aggressive towards, and try to expose, wannarexics in order to remove them from the community (Boero and Pascoe, 2012).

2.1.3. Pro-ED Content and Thinspo

The content posted in pro-ED communities is mostly content that can be viewed as pro-ED content. That is, content that in some form promotes EDs and disordered eating behaviors. This type of content is often either sharing tips and techniques for how to lose weight, fast or purge, or it can be so-called *thinspiration* (from thin-inspiration, also often referred to as *thinspo*) (Borzekowski et al., 2010).

Thinspiration is content that glamorizes very thin bodies, and can be in the form of images, text, video or audio/music. The purpose of this content is to motivate the members of the community to keep up their disordered eating habits. Most common are images of dangerously thin bodies, very often models or other famous people with eating disorders.

A contrast to thinspiration is *reverse thinspiration*, commonly referred to as reverse thinspo. This type of ED motivation usually comes in the form of photos. While thinspo depicts dangerously thin bodies, reverse thinspo depicts dangerously overweight bodies. The purpose of reverse thinspo is usually to scare the community members so that they will be motivated to keep up their disordered lifestyles.

2.1.4. Pro-Recovery

Pro-recovery communities is a counter-movement to pro-ED communities which focuses on and promotes recovery from EDs. The communities can consist of people suffering from EDs that wish to be healthy and break out of the eating disorder mentality, or of concerned parents, siblings or spouses. Even though the pro-ED and pro-recovery communities have different views on EDs, the two communities function in a similar way. Just as the members of a pro-ED community support and encourage each other, so do the members of pro-recovery communities. People share their struggle with their EDs and receive encouragement and motivation in return.

2.2. Big 5 Personality Traits

The Big 5 Personality Traits is a personality measure that scores the personalities of people based on five principles: openness, conscientiousness, extroversion, agreeableness, and neuroticism. The model is also sometimes referred to by other names such as the five-factor model, or the OCEAN model (from the acronym formed by the traits).

Since 1961, when Ernest C. Tupes and Raymond E. Christal first introduced a five-factor model to describe personality traits (Tupes and Christal, 1961), a lot of research has been done to mold and support the categorization model. Since 1980s and 1990s the model has been widely used in research and become one of the most well-regarded personality models (McCrae and John, 1992).

As described by McCrae and John (1992) and Costa and McCrae (2008), the personality found through the Big 5 personality model is based on a score of each of these five traits:

- **Openness to experience:** Fantasy, aesthetics, feelings, actions, ideas, values. Individuals with high openness scores are described as being generally open to experiences and ideas. They have a vivid imagination and are highly responsive to beauty. Their feelings are very important to them, and they have a moderate level of intellectual curiosity and liberal views.
- **Conscientiousness:** Competence, order, dutifulness, achievement striving, self-discipline, and deliberation. People with high conscientiousness score are rational and sensible when making decisions. They are described to be moderately neat, punctual and organized, but sometimes less dependable. They strive for excellence

in anything they do and have high aspirations, but tend to quit when things get too difficult.

- **Extroversion:** Warmth, gregariousness, assertiveness, activity, excitement seeking and positive emotions. Extroverted people usually enjoy big crowds and are seen as dominant and forceful. They have a high level of energy and frequently experience strong feelings of happiness and joy.

- **Agreeableness:** Trust, straightforwardness, altruism, compliance, modesty, and tender mindedness. People with a high agreeableness score tend to be friendly and compliant. They also show high levels of altruism.

- **Neuroticism:** Anxiety, angry hostility, depression, self-consciousness, impulsiveness, and vulnerability. Individuals with high neuroticism score are typically perceived as anxious people that struggle with feelings of frustration, irritability, and anger. Still, they only occasionally experience periods of unhappiness, just like what most people experience.

One common method to calculate the Big 5 personality traits of a person is to give the person a set of questions to answer. The number of questions used can vary, but is often around 40 - 50. The questions can be answered on a scale from strongly disagree to strongly agree where each possibility on the scale is weighted with a number (such as 1-5). The sum of the weighted numbers is what the trait scores are based on. It is the combination of these five trait scores that make the personality.

2.3. Twitter

Twitter is an online microblogging service with 335 million monthly active users (Twitter, 2018a). The word microblog comes from the term blog, an online piece of text usually intended to share the thoughts of the author with the world. A microblog is a shorter version of a blog, allowing the user to share short text updates (Passant et al., 2008). Microblogging services are websites solely dedicated to the sharing of these microblogs, and they usually set restrictions on the length of the texts to keep them short. On Twitter, a microblog-entry is called a tweet and it has a maximum character limit of 280. The tweet may contain not only text but also up to four photos, a GIF (short sequence of images) or a video (Twitter, 2018b). Links can also be included in the text. Figure 2.1 is an example of a typical tweet and the different elements it consists of. The example tweet contains both text and a link to a video.

Each user on Twitter has to create an account. In order to do this, the user has to enter a unique username and an author name. The author name does not need to be unique and it does not need to match the username. After having created the account, the user is given his/her own profile page and a news feed. The profile page contains, among other

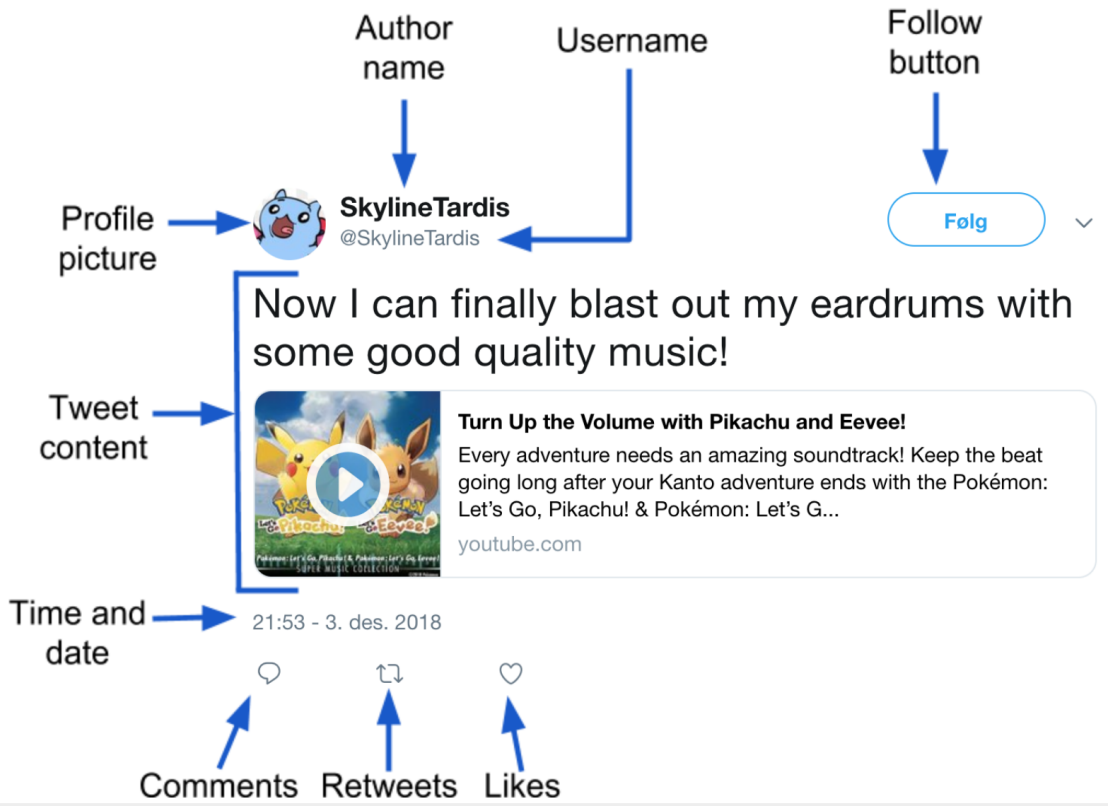

Figure 2.1.: Example of a Tweet

things, information about the user as well as a list of all the tweets the user has posted. Figure 2.2 shows an example of a Twitter profile page.

A Twitter user can choose to follow other Twitter users. If user X decided to follow user Y, then all the new tweets posted by user Y will show up on the news feed of user X. As seen in figure 2.2, the profile page also contains information about how many accounts a user follows and how many accounts are following the user. This is often a reference to the popularity of the user, where a high number of followers indicates a high level of popularity. A user can decide whether they want their tweet to be public (visible for all) or private (visible only to followers). Users are also able to like or retweet other tweets. Retweeting a tweet means that user X can share a tweet made by user Y on their own profile page while still crediting the original author, in this case user Y. It is also possible for a user to leave a comment on another user's tweet. This can create a long chain of comments where the users discuss, make jokes or comment on topics.

Communication on Twitter is performed through tweeting, retweeting, and leaving comments. Many users also include hashtags in their tweets. A hashtag always begins with the # (pound) symbol, followed by a word or a merged together sentence. One way

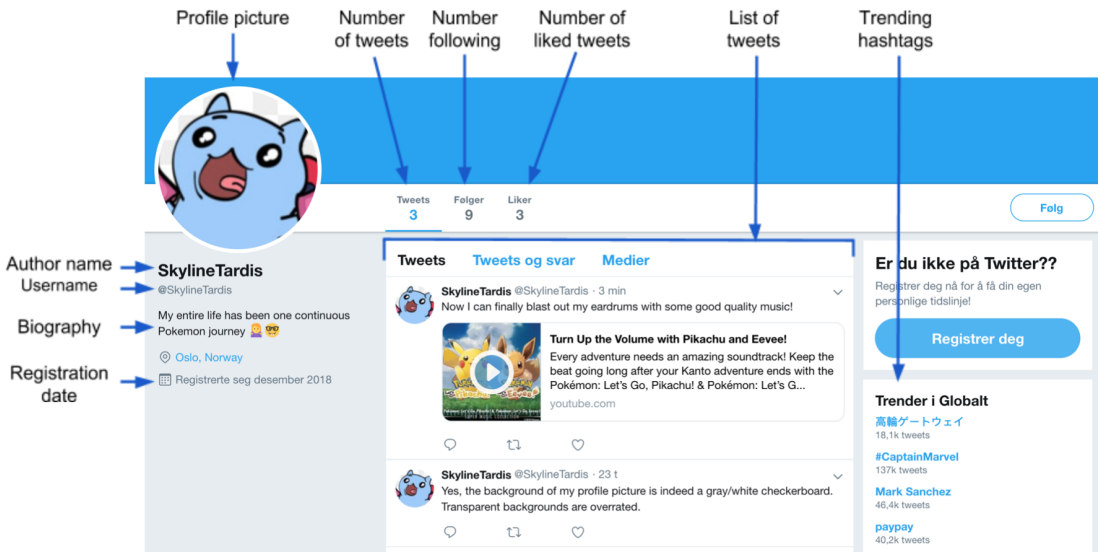

Figure 2.2.: Example of a Twitter Profile Page

of finding new users to follow is to search for specific hashtags. It is common practice to include hashtags in tweets to explain what the tweet is about. Some examples of hashtags being used by the pro-ED community are #proAna, #skinny4christmas, and #thinspo. #proAna usually refers to the person posting the tweet being pro-anorexia, #skinny4christmas is used to mark a tweet that contains information about the yearly pro-ED event where people compete to be skinny by Christmas, and #thinspo refers to a tweet containing thin-inspiration content.

3. Machine Learning for Text Classification

This chapter is a continuation of the previous chapter, where the information and theories needed in order to understand the content of the thesis is explained. The focus of this chapter is terms related to machine learning for text classification. The technical concepts surrounding machine learning are presented first, followed by the machine learning algorithms for text classification used in this thesis. The chapter ends with an explanation of the different classification metrics used to measure the accuracy of the machine learning algorithms.

3.1. Machine Learning Concepts

Some common elements in many machine learning algorithms are described in this subsection. First, kernel functions are described, which are used among others in Support Vector Machine (SVM) algorithms and Gaussian Process (GP) algorithms, which are described in chapter 3.2. The term overfitting is then explained before elaborating on cross-validation, which is a way to avoid overfitting the model to the training data. Lastly, stop word removal is explained.

Kernel Functions
Kernels functions are used in machine learning algorithms like SVM and GP. The kernel functions are ways of representing data as a numerical vector that represents potentially relevant features (some of the most common features to use in natural language processing are described in chapter 4.1). The main benefit of kernel functions comes from the *kernel trick*. The kernel trick makes it possible to compute the inner product of each pair of mapped points instead of computing this high-dimensional mapping explicitly:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \qquad x, x' \in \mathbb{R}^d \tag{3.1}$$

where $\phi : \Re^d \to \mathcal{H}$ is the feature map, d is the dimension of x, $\mathcal{H}$ a Hilbert space and k the kernel function such that $k : \mathbb{R}^d \text{ x } \mathbb{R}^d \to \mathbb{R}$.

Another property of kernel functions that makes them a useful tool is that it is possible to combine several kernels without losing performance. This means that one can create kernels for specific features of a dataset and combine them to get a more accurate classification than what would be possible with one kernel function alone. Many different

functions can be used as kernels, and two common examples are the Radial Basis Function (RBF) kernel and the linear kernel. Both of these kernels are commonly used in machine learning algorithms such as Support Vector Machine (SVM). What makes one kernel different from another is how it transforms the input data into a higher dimensional space. The linear kernel is popular because it is often very quick, whilst the RBF kernel often gives more accurate results.

Overfitting
Overfitting is a common problem in supervised machine learning. This problem occurs when an algorithm has become too well fitted to the training data, and thereby performs worse on test data. Another way to see it is that the algorithm has *learned* the noise in the data as well as the meaningful parts. This is more likely to happen to more complex algorithms, because these have a greater possibility of tuning their parameters to fit the data. When a model is overfitted, it will not be able to generalize well, and thereby not perform well when given unseen data.

Cross-Validation
To avoid overfitting issues, cross-validation (CV) can be used. In CV, a small subset of the training data is held back while the classifier is trained on the remainder. After training, the classifier is then tested on the portion of the training set that was held back. Normally, a method called k-fold CV is used. Here the data is split into k equal sized folds and trained on k-1 folds, the last fold is used for testing. This is then done with each fold held back so that all the data is still used in training and therefore it does not waste any training data while still being able to counter the problem of overfitting.

Stop Word Removal
When dealing with natural language processing tasks, it is common to remove so-called stop words. Stop words are words that are so commonly used in a language that they no longer have any distinguishing power when it comes to seeing the difference between two documents. Examples of these can be *is*, *to*, *this* and *be*. To avoid spending computational power on trying to classify these words they are often removed in pre-processing. This means that the sentence:

> Eating disorders are serious illnesses that can cause serious health problems

Would be reduced to:

> Eating disorders serious illnesses cause serious health problems

The number of words, and which to remove, varies depending on the language, the problem, the dataset and the machine learning algorithms that are used.

3.2. Algorithms

Machine learning has been a popular topic for many years and a number of different machine learning algorithms exist today. Different methods have been developed for different kinds of tasks, and in the case of this thesis, models that were good for Natural Language Processing were the most relevant. The machine learning algorithms that were used in this thesis were the Support Vector Machine, Gaussian Process, K-Nearest Neighbors, Ridge Regression and Multilayer Perceptron.

3.2.1. Support Vector Machines

The Support Vector Machine (SVM) algorithm is a machine learning algorithm that has been shown to give very good results in text classification tasks (see chapter 5.2 and 5.5). The main goal of an SVM is to map data onto a higher dimensional space so that it can be able to find a hyperplane that will separate the data into the correct classes. A hyperplane is a subspace of its ambient space (space that surrounds an object). If a space is 3-dimensional, then the hyperplanes of that space are the 2-dimensional planes that cut through it. The SVM algorithm attempts to find the hyperplane with the largest possible margin from the closest points of each class. The SVM finds this hyperplane by applying a kernel function to the data, which maps the data points onto a higher dimension. By doing this, it can find a way to linearly separate data that is not originally linearly separable. The kernel function uses the feature vectors of the original space as input and output the dot product of the data in the feature space. This makes it a lot less computationally expensive than the alternative of elevating the whole dataset to a higher dimension to compute the dot product from the transformed data.

Figure 3.1 shows a dataset that has been separated into two classes by three different hyperplanes presented as black lines. The grey areas on each side of the hyperplanes are the margins which go out to the closest support vector. In this instance, it is clear that one of the hyperplanes gets a lot larger margins than the other two, even though all three hyperplanes are successful in splitting the data. This can be seen by the larger gray area around the hyperplane. The goal of the SVM algorithm would be to find the hyperplane that has this property.

Sequential Minimal Optimization
Sequential Minimal Optimization (SMO) is an iterative algorithm used in the training of a SVM. The algorithm was proposed in 1998 by Microsoft Research and was created to solve a problem which SVMs struggle with (Platt, 1998). The problem arises during training of the SVM and involves having to solve a large quadratic programming (QP) optimization problem. SMO solves this by dividing the problem into a series of the smallest possible sub-problems which it then solves analytically. While SVM use numerical QP as an inner loop, SMO instead uses an analytic QP step. As a result, SMO avoids using the time-consuming loop and thereby speeds up the training process. SMO uses only a linear amount of memory when it comes to the training set size. This means that SMO can

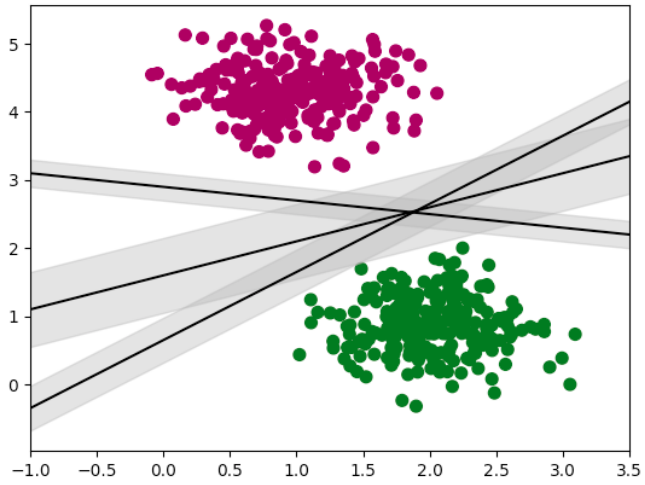

Figure 3.1.: Example of a Dataset Separated by Possible Hyperplanes

handle very large training sets and has better scaling properties than the standard SVM training algorithm.

3.2.2. Gaussian Processes

Gaussian Process (GP) is, just as SVM, a kernel-based approach to machine learning. It takes a non-parametric and probabilistic approach, which is well suited for regression in language processing and data mining. In order to define GP, it is useful to first know what a random process is. A random process is a collection of random variables over a common probability space. For the random process $\chi \to \mathbb{R}$ there will be one random variable for each $x \in \chi$. The distribution of a random variable one gets from evaluating the process at some finite point, $x \in \chi$, is called finite-dimensional distribution. In a GP, this finite-dimensional distribution is multivariate Gaussian. The main idea behind GPs is that if x_1 and x_2 are close together in the input space, they are probably also close together in the output space, that is $p(f(x_1), ..., f(x_N))$ follows some Gaussian distribution. GP is defined by a mean function and the covariance function to output the expected value of *f(x)*. The mean function $m(x)$ is defined by:

$$m(x) = \mathbb{E}[f(x)] \tag{3.2}$$

The covariance function K is defined by:

$$K(x_1, x_2) = \mathbb{E}[(f(x_1) - m(x_1))(f(x_2) - m(x_1)] \tag{3.3}$$

K returns a measure for the similarity between x_1 and x_2, as well as how similar $f(x_1)$ and $f(x_2)$ should be. Two properties have to hold for K, namely that it has to be *symmetric*

and *positive semi-definite*. These are the same restrictions that hold for a kernel, which is why this is often also referred to as the kernel function for the GP.

One of the things that make GP a good choice for text classification and regression is that it allows for explicit quantification of noise and a modulation of features by fitting a kernel function, the covariance function, to the known data. Because it is possible to choose the most suitable kernel function for the problem at hand, the GP model is flexible and can be used for many different problems with good results. GP has been found to be very effective for short text classification when combined with word embedding (Ma et al., 2015).

3.2.3. K-Nearest Neighbors

K-Nearest Neighbors (k-NN) is a machine learning algorithm that uses training vectors and pattern recognition to label a data element. The training vectors are used to find a given amount of elements (k) close to a selected data element, hence the name K-Nearest Neighbors. The data element will get a label based on the labels of the nearest neighbors. The labeling choice is usually done by a simple majority vote. A common way of measuring the distance between a data element and its neighbors is by using the Euclidean distance measurement. The Euclidean distance between two points ($\mathbf{p}$ and $\mathbf{q}$) is the length of the line segment connecting them. In Cartesian coordinates, if $\mathbf{p} = (p_1, p_2, ..., p_n)$ and $\mathbf{q} = (q_1, q_2, ..., q_n)$ are two points in Euclidean n-space, then the distance (d) from $\mathbf{p}$ to $\mathbf{q}$, or from $\mathbf{q}$ to $\mathbf{p}$, is given by the Pythagorean formula:

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \quad (3.4)$$

Another method of measuring distance is by using the Hamming distance. This measure is usually employed when using the k-NN algorithm for text classification. The distance between two text elements is the minimum number of substitutions required to change one string into the other, or the minimum number of errors that could have transformed one string into the other. The Hamming distance between the words *proana* and *promia* is equal to two because of the substitution of either *an* in *pro**an**a* to make it identical to *promia*, or the substitution of *mi* in *pro**mi**a* to make it identical to *proana*.

Figure 3.2 displays an example of how classification with the k-NN algorithm can be done. In the example, there are two labels: unrelated (U) and pro-ED (P). The blue circle in the middle is a Twitter account that is going to be categorized into one of the two labels. The algorithm has been trained by using a dataset consisting of Twitter accounts already labeled. As seen in the example, the Twitter accounts have been separated into two groups, a green group consisting of pro-ED labeled accounts and a red group consisting of unrelated labeled accounts. The algorithm calculates the similarity between the blue account and the two groups of labeled Twitter accounts and places the blue account

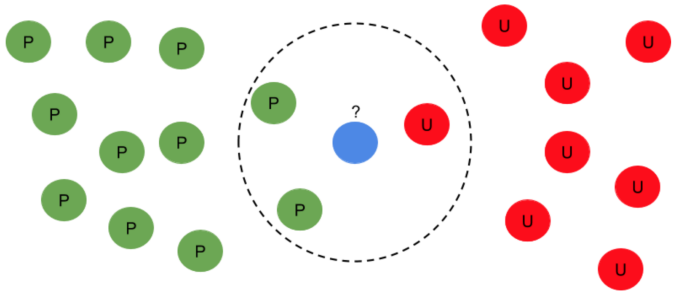

Figure 3.2.: Example of K-Nearest Neighbors Classification

somewhere between the two groups. In order to find the correct label for the blue Twitter account, a number of neighbors has to be specified. In this case, the number is three (represented by the black dotted line). The three nearest neighbors to the blue Twitter account is then calculated based on the distance. One of the neighbors belongs to the unrelated group while two of the neighbors belong to the pro-ED group. Because a majority of the neighbors are labeled as pro-ED, the blue account will get this label as well.

The main advantage of the k-NN algorithm is that it can be used both for regression and for classification.

3.2.4. Ridge Regression

Ridge Regression (RR) is an algorithm intended for regression where the output value is expected to be a linear combination of the input variables. RR is often beneficial because it deals with the multicollinearity problem that often arises in regular linear regression. Multicollinearity is the existence of near-linear relationships among the independent variables. For example, if there are three variables with a perfect linear relationship to begin with, then during regression their relationship will cause a division by zero. This would normally cause the division to abort. Ridge regression solves this problem by introducing a weight penalty on the size of the coefficients. When this penalty is added to the equation, the relationship between the input variables is not exact, and therefore the division by zero will be avoided.

Because it solves the problem with multicollinearity, RR is often a good choice when the input variables are strongly correlated. It is also good at handling cases when the number of features is large compared to the number of observations. RR is also a very good approach when dealing with Natural Language Processing tasks where the input is often highly dimensional. With more complex models, it is hard to avoid overfitting when the number of input variables is high. Being a linear, and thereby less complex, approach, RR is less likely to overfit to the training data in these situations.

3.2.5. Multilayer Perceptron

A Multilayer Perceptron (MLP) algorithm is a feed-forward neural network consisting of at least three layers of nodes: an input layer, a hidden layer, and an output layer. While there can be only one input and output layer, there is no limit to the number of hidden layers that the algorithm can have. Each of the layers consists of nodes called perceptrons, which is the simplest form of an artificial neural network. The network is trained using backpropagation, a supervised learning technique which, as the name suggests, propagates changes to the weights backward through the network. Both linear and non-linear activation functions can be used for the neurons in the hidden layers. The input can be of any size and the output can be both classification and regression predictions. The output of a node in the network is defined by an activation function. This function tells the node what to output based on the input it gets. There are many options for this activation function, but the activation function used in this thesis is the rectifier linear unit (ReLU), which is calculated by the formula:

$$f(x) = x^+ = \max(0, x) \tag{3.5}$$

Each node in a layer connects with a weight, w_{ij}, to every node in the following layer. The MLP algorithm is trained by changing the weights after processing a piece of data. The changes to the weight are based on the number of errors in the output compared to the expected result. The error in output node j in the nth data point can be represented as $e_j(n) = d_j(n) - y_j(n)$, where d is the goal value and y is the value produced by the perceptron node. The weights are adjusted based on corrections that minimize the error of the entire output. These corrections are given by:

$$\mathcal{E}(n) = \frac{1}{2}\sum_j e_j^2(n) \tag{3.6}$$

Gradient descent, an optimization algorithm for finding the minimum of a function, is then used in order to find the change in each weight:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n) \tag{3.7}$$

y_i is the output of the previous node and η is the learning rate (to what extent new information overrides old information). v_j is an induced local field.

3.3. Classification Metrics

To evaluate how good a model performs, there needs to be some form of metrics that measures the accuracy of the results it produces. In this section, the measurement metrics used in this thesis are described. These metrics are precision, recall, F_1 score and Pearson correlation coefficient.

Table 3.1.: Explanation of Retrieval Accuracy

		predicted	
		negative	positive
actual	negative	true negative	false positive
	positive	false negative	true positive

Precision and Recall
Precision and recall are two very important concepts to consider when it comes to evaluating the accuracy of a text classification model. Precision describes the degree to which the model is able to retrieve true positives, meaning how many of the elements in the result that were actually positive, compared to how many were labeled positive. This is computed by the formula:

$$P = \frac{Tp}{Tp + Fp} \tag{3.8}$$

Where Tp in the is short for true positive, Fp is short for false positive.

Recall, on the other hand, describes how many of the relevant documents were actually retrieved. This means that it describes how many of the actual positives were in fact identified and labeled as positive. Recall is calculated by:

$$R = \frac{Tp}{Tp + Fn} \tag{3.9}$$

Fn is short for false negative. The relationship between true and false negatives and positives can be seen in table 3.1.

Both of these metrics have their advantages and disadvantages. Precision has the advantage that it describes how good the model is at finding true positives, but the disadvantage that it does not take into account how many of the actual positives were labeled as negative. With recall, however, it is described how many of the relevant documents were retrieved, but not how many actual negatives were labeled as positive. Because of this, using either of these measures alone results in an incomplete metric to evaluate the accuracy of the model. This is why they are mostly used together, often in combination with F_1 score.

F1 score
The F_1 score, often referred to as the F-score or F-measure, is a way of combining precision and recall to get a better evaluation of the accuracy of the machine learning model. The F_1 score is a representation of the weighted average of precision and recall, and is calculated by the formula:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{3.10}$$

The best possible F_1 score is 1, whilst the worst is 0. F_1 score is often considered a more robust metric for calculating accuracy as it balances out the results of the precision and recall metrics. It is a good way of filtering out extreme values for either precision or recall.

When calculating precision, recall and F_1 scores, it will always be with respect to one label. When only two labels are present, it is often adequate to calculate the score based on the *positive* label. This is also called a binary calculation. However, for multi-class problems, the calculations have to be done a bit differently. The options of how to calculate the scores are:

- **Micro:** Micro calculations means that all the true positives, false positives, true negatives and false negatives are calculated globally for the entire dataset.
- **Macro:** Macro calculations means that an unweighted average of the scores for each label is calculated.
- **Weighted:** The weighted approach does the same as the macro calculations, except the average is calculated with weights for each label that represent the size of the label. This is often a good approach when the classes are uneven.
- **Sample:** The sample setting will calculate the average of the scores for each instance.

Which approach is best is dependent on the task at hand, and the different approaches will often yield very different results.

Pearson Correlation Coefficient
The Pearson correlation coefficient is a measurement used in statistics to describe how two datasets are linearly correlated. The coefficient can take on values from -1 to 1, where both -1 and 1 mean that they are perfectly linearly correlated. -1 being in opposite directions, 1 being in the same direction, and 0 meaning there is no correlation at all. As it can show the linear relationship between the true values and the values predicted by the regression model, the Pearson correlation coefficient is one of many metrics to evaluate the fit of a regression.

4. Text Representation, Annotation, and Tools

As the last of the three chapters, where the information and theories needed in order to understand the content of the thesis are explained, this chapter covers terms and concepts from three different topics. The chapter starts by introducing the first topic, which is the most common text representation models used in Natural Language Processing (NLP). The second topic introduces the process of doing data annotation. Finally, the last topic presents the tools used in the research phase of the thesis.

4.1. Text Representation Models

One of the main challenges when dealing with data in the form of text, is text representation - how to numerically represent the text. In order to use the machine learning models described in the previous chapter, the data first needs to be represented in a way that the model can interpret. This is what text representation does. The text representation models (also called features) used in this thesis are presented below.

4.1.1. Word Embedding

Word embedding is one alternative for encoding words into vectors. Vector encoding is a collective term used to describe techniques where words are converted into real numbers and represented as vectors. This way of representing text allows words with similar meaning to have a similar representation, thereby capturing their meaning. A very simple form for vector representation can be described as follows: Say there exists a sentence like *Eating disorders affect a large part of the population*. It is possible to create a dictionary consisting of all the unique words in this sentence. The dictionary might look like this:

['Eating', 'disorders', 'affect', 'a', 'large', 'part', 'of', 'the', 'population']

To represent a word in this sentence as a vector one can encode the word in such a way that a 1 represents the position in the dictionary where the word is located and 0 otherwise. The vector for the word *affect* will by this method look like this:

[0, 0, 1, 0, 0, 0, 0, 0, 0]

Table 4.1.: Example of Three Different N-Grams

Phrase	1-gram	2-gram	3-gram
to be or not to be	to, be, or, not, to, be	to be, be or, or not, not to, to be	to be or, be or not, or not to, not to be

Word embedding is a very popular way of vectorizing the words in a document. In general, word embedding helps learning algorithms achieve better results on NLP tasks by bringing similar words closer together (Mikolov et al., 2013). It has also been shown to improve learning methods dealing with short text (Kenter and De Rijke, 2015).

4.1.2. N-gram

An n-gram is a sequence of words or characters that appear next to each other in a text document. The number of words or characters in the sequence is decided by n. If n=1 the 1-gram is called a unigram, while if n=2 it is called a bigram and so on. Which n-value is optimal will depend on the task the n-gram is trying to solve. When n is big, it is possible to store more context than with a small n. Table 4.1 contains an example of three n-grams for the phrase *to be or not to be*.

As seen in table 4.1, the 1-gram (unigram) divides the text into separate words, while the 2-gram (bigram) pairs the first two words together followed by the pairing of the next two words. The 3-gram (trigram) pairs the three first words and so on. One of the main advantages of using n-grams is that it, in combination with probabilities, is possible to find out how often certain words appear together. For example, the bigram in table 4.1 that looks at two and two adjacent words, will be able to detect that the word *to* is often followed by the word *be*. This can be used for numerous things in NLP, like predicting the next word, spelling corrections, and sentiment analysis.

Another advantage with n-grams is that it can be used on a character level, not only on word level. This means that it is possible to detect characters that appear together in the same way as detecting words that appear together.

4.1.3. Bag of Words

Bag of Words (BoW) is a representation of words that machine learning algorithms can process. In a BoW model, all the words are stored in the model, but the order is ignored. This means that a BoW model is, in fact, the same as an n-gram model when n=1. The model essentially consists of only two things: a collection of known words and a count of how many times each of these words appear in the text. BoW can be made more or less complex by choosing to include or ignore text features such as case, numbers, and punctuation. As it produces a vector representation of the words in the document, it

Table 4.2.: Example of the Mechanism of a Bag of Words Model

Term	Doc1	Doc2
give	1	0
recovery	1	1
hard	1	0
best	0	1
friend	0	1

can also be used as input to many other models. Topic models (see chapter 4.1.6), for example, can use the BoW model as input to find more complex structures in the text. As an example of how BoW can be used to represent the words in two different documents, consider the following two documents:

Doc1: I give up, recovery is hard.
Doc2: My best friend is in recovery.

The words in table 4.2 contribute to the meaning of the text in doc1 and doc2. Each unique word gets a value based on the presence in one of the documents. The first word *give* is present in only the first document, so it gets the value 1 for Doc1, and 0 for Doc2. Since the second word *recovery* is present in both documents, it gets the value 1 for both documents, indicating that it is present. The rest of the words are all present in just one of the documents and therefore get a 1 for the document they are present in, and a 0 for the other document.

4.1.4. Part of Speech

Part of Speech (POS) tagging is a way of explaining how a word is used in a sentence by assigning a tag to each word. A word can have one of several tags, which are normally found from a dictionary or a morphological analysis. Using statistical models, a sequence of POS tags can be drawn from a sequence of words. This is usually done by using hidden Markov models, a statistical model which looks at the current word in the context of the surrounding words in order to make a more accurate prediction. A sliding window is often used in the analysis, looking at features of the surrounding words (also called the context) of the word that is being tagged. The POS tags of the surrounding words, that have already been tagged, are also used as features to determine the tag(s) of the word in question.

The number of tags that are used depend on the *tagger*, or model, that is being applied. The tags are often given from a lexicon or a finished model. These can contain anywhere from eight tags, being the general word classes (noun, verb, pronoun, preposition, adverb, conjunction, adjective, and article), to several hundreds. Which tagger is the best depends on the problem at hand and how detailed information is required.

Table 4.3.: Example of Part of Speech Analysis of a Sentence

Good	Adjective
morning	Noun
world	Noun
lets	Verb
continue	Verb
to	Particle
prosper	Verb

An example of how POS tagging works can be seen below:

Good morning world lets continue to prosper

A simple POS tagger online[1] produces the results that are shown in table 4.3. This specific tagger is based on the POStagger from Stanford University which is also available for download as a java program[2]. The table shows how the tagger gives tags to different words depending on how they are used in the sentence. In the example, the tagger correctly finds the word class of each word, which can be used to describe the linguistic style of the author.

4.1.5. Term Frequency-Inverse Document Frequency

Term Frequency-Inverse Document Frequency (TF-IDF) is a numeric measurement which reflects the importance of a word that appears in a document or a text. TF-IDF is calculated by adding together the two measurements: Term Frequency (TF) and Inverse Document Frequency (IDF):

$$\text{TF-IDF} = TF * IDF \tag{4.1}$$

TF is a measurement of the number of times a word appears in a document. If a word has a high number of appearances in a document, then the word gets a high value. In contrast, if a word appears few times in a document, the word gets a low value. One area where TF is much used is information retrieval, for example in search engines. When searching for a document online one usually enters a few words into the search engine which describe what it is one is searching for. If one searches for eating disorder related documents, then it is likely that a document containing multiple mentions of the words *eating* and *disorder* (high TF values) is relevant to the search. If a document contains no mention of either *eating* or *disorder* (low TF values) then it is safe to say that the document is irrelevant.

[1] `https://parts-of-speech.info/`
[2] `https://nlp.stanford.edu/software/tagger.shtml`

IDF, on the other hand, is a measurement of the number of documents that mention a word. The higher the IDF value, the less unique the word is assumed to be. IDF can be calculated by using the following equation:

$$IDF = \log(\frac{N}{n}) \tag{4.2}$$

N is the total number of documents and n is the number of documents that a word has appeared in.

IDF is important because it gives more power to words that are mentioned fewer times in a set of documents. Thereby it can be used to achieve a more accurate retrieval result in a document search. If a set of 20 documents all contain the term *coffee*, then it will be difficult to know which of the documents are of most relevance. If instead, only two documents contain the word coffee, then it is much easier to return relevant documents. In other words: if a word has appeared in all the documents, then that word is probably not relevant to a particular document. But if it has appeared in a smaller subset of documents then the word is likely to be of some relevance to the documents it is present in.

4.1.6. Topic Modeling

Topic Modeling is a method for analyzing large amounts of unlabeled text in order to obtain groups of words that describe the information in the text. A *topic* is a cluster of words that frequently occur together, and by adding contextual information the models can also detect when the same word is used in different ways. Topic modeling can be useful in discovering the topic of the document at hand, and also give a measure of how much of the document belongs to each topic if several topics are present in the document. It can also discover *hidden* topics, meaning semantic structures that are not obvious to a human, that are present in the text. Topic modeling can be used to classify documents based on their topics, so as to organize large corpora of data. Topic modeling can also be useful in text classification and annotation, which is the objective of this thesis.

Obtaining the models can be done using many different techniques, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003), Vector Space Model, Latent Semantic Indexing and Probabilistic Latent Semantic Analysis. The most popular method is LDA, which is a matrix factorization technique where each document is made up of several topics. LDA works as follows: The algorithm first assigns a random topic t to each word w. The amount of topics included is decided beforehand as a parameter when building the model. Then, going through each document d, the algorithm determines $P1 = P(t|d)$, and $P2 = P(w|t)$ for each word in the document. Finally, the given word is updated with the probability $P1 * P2$ for the assignment it was given. This process continues for a given number of iterations or until the algorithm converges.

4.1.7. Non-Linguistic Features

Several non-linguistic features can also be used in the classification of social media users and text. Non-linguistic features are features that are not drawn from the linguistic properties of the text, but rather from information about the author. Examples of non-linguistic features that can be relevant to look at in a Twitter account can be age, gender, the structure of their network (followers and followings), and the number of tweets. Behavior can also be used as a feature, such as how often the user tweets, use of hashtags, images and retweets, and how often other people like or retweet their tweets. These features can often be extracted from account metadata.

Even though the use of non-linguistic data is limited in this thesis, the study of related research, presented in chapter 5, shows that non-linguistic features are used in several studies with good results. One example is Giæver (2018), who used Twitter account usernames to get predictions as to whether a Twitter account was pro-ED or not. Non-linguistic features have also been used by Kumar et al. (2017) and Solomon et al. (2019) to predict personality in social media, both found these features to be useful in the predictions.

4.2. Annotation

In order to train a machine learning algorithm, a labeled dataset is needed. When labeling a dataset there are several ways to figure out the labels for each data element. Sometimes the labels can be found from ground truths in the data itself, but in the cases when no ground truth is provided explicitly in the data, the labels have to be found through data analysis. This analysis can be done automatically by machines or manually by humans.

4.2.1. Automatic Annotation

When there is a lot of data to analyze, or the data is too complicated for humans to understand, automatic annotation can be a good alternative. Automatic annotation involves a computer program looking at the data to determine the label that is most likely to be correct. When applying this type of annotation, there will often be some degree of error which should be taken into consideration when evaluating the results. Another drawback to automatic annotation is that there first needs to be some previously annotated data for the machine to use for training. This means that automatic annotation is only possible after a certain amount of annotated data is already gathered. The accuracy of the automatic annotation will then become dependent on the similarity between the training data and the data to be annotated. It is also dependent on the amount of training data available as well as the complexity of the problem.

4.2.2. Manual Annotation

Manual annotation is a data labeling method that involves human annotators looking at the data in order to determine which label it belongs to. To evaluate whether a Twitter account is pro-ED or unrelated requires information that is not already explicitly specified in the account data. The human annotators have to evaluate the content relevant to the Twitter account, such as tweets, in order to see if there is an indication in the content that reveals what is the correct label for the Twitter account. This takes time, which is why manual annotation usually is done only in order to train a machine to do it automatically. When a machine is thoroughly trained and provides satisfying results, then the need for manual annotation diminishes but until then, manual annotation is a good way to measure the accuracy of the automatic annotation.

To reduce mistakes and bias, the manual annotation is preferably done by multiple annotators. This brings a need for a measurement of the agreement of the annotators. Inter-annotator agreement can be measured in various ways, and two common metrics are Cohen's kappa (Cohen, 1960) and Fleiss' kappa (Fleiss et al., 2013). It can be assumed that when inter-annotator agreement is high between multiple annotators, the annotations are more likely to be correct.

Cohen's Kappa
Cohen's kappa (κ) was introduced by Cohen (1960), and since it was the first introduction of the kappa coefficient, it is often referred to as just *kappa* (κ). Cohen's kappa is designed to determine the agreement for nominal scales and takes values between -1.00 and 1.00. The coefficient takes into account the probability of agreement, and therefore becomes more robust than simple percentage calculations of agreement. To calculate κ, the following formula is used:

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \tag{4.3}$$

Where p_0 is the proportion of agreement between the annotators and p_e is the proportion of agreement that would be expected by chance.

Cohen's kappa has become popular for agreement measuring because of its simplicity and robustness. It does, however, have some drawbacks in that all types of disagreement is treated the same. This could be a problem if there are many classes and some are closer to each other than others. Another limitation is that Cohen's kappa only measures agreement between two annotators. To solve these issues, several variants of the kappa measure has been proposed over the years. These variants build upon the original Cohen's kappa in order to try to improve on the limitations. One of them is Fleiss' kappa.

Fleiss' Kappa
About 20 years after Cohen proposed his kappa, an alternative variation of the measure was proposed by Fleiss et al. (2013). This version of the kappa allows for any fixed number of annotators, and thereby is more suited for cases with more annotators. This, however,

comes with the drawback that it requires adding some complexity to the calculations. The first thing that needs to be calculated is the proportion of all data elements that were assigned to the j-th label, (p_j). This can be done with:

$$p_j = \frac{1}{Nn} \sum_{i=1}^{N} n_{ij} \tag{4.4}$$

Where N is the total number of elements to classify into labels, n is the number of annotations per element, and n_{ij} is the number of annotators who assigned the i-th element to the j-th label.

The second thing that needs to be calculated is p_i. p_i is a measure of how much the annotators agree on a label for the i-th data element. This is calculated by:

$$p_i = \frac{1}{n(n-1)} \sum_{j=1}^{k} n_{ij}^2 - n_{ij} \tag{4.5}$$

k is the total number of labels. Finally, p_0 and p_e can be calculated by:

$$p_0 = \frac{1}{N} \sum_{i=1}^{N} p_i \tag{4.6}$$

$$p_e = \sum_{j=1}^{k} p_j^2 \tag{4.7}$$

These p_0 and p_e values are equivalent to the p_0 and p_e used in Cohen's kappa formula and can therefore be inserted into the Cohen's Kappa formula in order to get the measured agreement.

4.3. Tools

The tools that were used in the research phase of this thesis are presented below. A number of different tools were considered, both for feature extraction and for building the machine learning classification models, but only those that ended up being used are described.

LIWC

Linguistic Inquiry and Word Count (LIWC)[3] (Pennebaker et al., 2001) is a program that can be used to analyze text. The goal of LIWC is to discover the percentages of the words in the text that reflect different features related to the social and psychological states of a person. These language categories can be in relation to emotions, styles of thinking, social concerns and even Part of Speech (POS) tags. LIWC is often used as a

[3] `https://liwc.wpengine.com/`

baseline to measure the performance of models, as can be seen in chapter 5.2, but the results can also be used as features used in classifiers.

SciKit-learn
SciKit-learn[4] (Pedregosa et al., 2011) is a popular Python library for working with machine learning. It contains simple and efficient tools for data analysis and machine learning, and it is built on top of NumPy, SciPy, and matplotlib. The library is open source, frequently updated, and has a large community of contributors and users. This library is also well documented and used in many guides on machine learning for Python. All of this makes it a very good tool for implementing machine learning models in Python.

Natural Language Toolkit
Natural Language Toolkit (NLTK)[5] (Bird et al., 2009) is a Python library that provides tools for many of the Natural Language Processing tasks relevant to this thesis. The library contains methods for most of the highly used text representation methods, such as n-grams and POS. NLTK is a free, open source library that is still maintained and frequently updated. It is well documented and one of the leading platforms for dealing with linguistic data in Python.

The GloVe Model
The Global Vectors (GloVe) model, is a machine learning algorithm that finds word embeddings and directly captures the global corpus statistics (Pennington et al., 2014). In this model the algorithm will create a co-occurrence matrix, X, where each element X_{ij} represents how often word j appears in the context of word i. A matrix of co-occurrence probabilities is also created, showing how likely word j is to appear in the context of word i. It then applies a cost function presented as a least squares problem using the equation:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \widetilde{w}_j + b_i + \widetilde{b}_j - \log X_{ij}) \tag{4.8}$$

V is the size of the vocabulary, $w \in \mathbb{R}^d$ are word vectors and b_i is the bias for word vector i. Pennington et al. (2014) states that for the weighting function, f, any function can be used as long as it satisfies the properties below:

1. $f(0) = 0$. If f is viewed as a continuous function, it should vanish as $x \to 0$ fast enough that the $\lim_x \to 0 f(x) log^2 x$ is finite.

2. $f(x)$ should be non-decreasing so that rare co-occurrences are not weighted too high.

3. $f(x)$ should be relatively small for large values of x, so that frequent co-occurrences are not given higher weights than appropriate.

[4] `https://scikit-learn.org/stable/`
[5] `https://www.nltk.org/`

GloVe also provides several pre-trained vectors[6] that can be applied without training a model. The options available are from Wikipedia, common crawls and Twitter. The Twitter vector set has been trained on 2 billion tweets, includes 27 billion tokens, has a vocabulary of 1.2 million words, and contains vectors of 50, 100, 150 and 200 dimensions.

Gensim
Gensim is another Python library made for NLP. It was created in 2008 by Řehůřek and Sojka (2010) with the goal of creating topic models from plain text. The library is free, open source and since its creation, it has been continuously improved upon to become an even better tool. Gensim is used in many Python examples of topic modeling and is cited in more than a thousand papers and student theses related to NLP, which makes it a good choice as a tool for topic modeling.

ARK Tweet NLP POS tagger
The ARK Tweet NLP Part of Speech Tagger[7] is a Java-based POS tagger created by Carnegie Mellon University that is designed to be used on tweets. The tagger is pre-trained on manually annotated POS tags for tweets, as well as hierarchical word clusters from unlabeled tweets. In this thesis, in order to be able to use the tagger with Python, a wrapper[8] was used. This wrapper sends tweets from Python to Java, where the operations are made, and returns the results back to Python.

Langdetect
Langdetect[9] is a Python library that is ported from Google's language-detection library[10]. The library is used for detecting different languages in text, and is very simple to use. To find the language of a piece of text, the text is given as an input to langdetect, which returns a label saying what language it is most likely to be written in. Langdetect includes support for 55 languages, but also supports adding new language profiles, making it a good tool for differentiating between many languages.

[6] https://nlp.stanford.edu/projects/glove/
[7] https://www.cs.cmu.edu/~ark/TweetNLP/
[8] https://github.com/ianozsvald/ark-tweet-nlp-python
[9] https://pypi.org/project/langdetect/
[10] https://code.google.com/archive/p/language-detection/

5. Related Work

This chapter contains an overview of what has been done in research related to this thesis. The chapter is split up into five parts. The first part tackles the important question of whether the use of social media is representative of real life. The remaining four parts cover each of its own related research area: personality detection through tweets, personality in eating disorder sufferers, online pro-ED communities and pro-ED detection on Twitter.

5.1. Social Media as a Representation of Reality

Kumar et al. (2017) asked in their paper on value and personality deducing in social network communities, a question which holds great importance when deciding to do research based on social media accounts. The question addresses whether social media is a good representation of the offline society. What makes this question important is that when doing research on social media, such as the personality of Twitter account owners, it is with an impression that the personality detected through the account is a sufficient representation of the personality of the account owner.

In research done by Back et al. (2010), it was found that the personality fronted on Facebook accounts is, in general, accurate to the offline personality of the account owner, and not a falsely reflected self-idealization. Golbeck et al. (2011a,b) support this finding by using machine learning algorithms, in combination with the Big 5 personality model, to accurately deduce personality from Facebook accounts and Twitter accounts. However, since that research did not look at pro-ED accounts, it is not a given that this is still accurate when it comes to the personality of pro-ED Twitter account owners.

Research has shown that a lot of pro-ED accounts are secondary accounts, created to solely focus on pro-ED content, whilst the user keeps a main account for communication with friends and family (Juarascio et al., 2010; Yeshua-Katz and Martins, 2013). These secondary accounts often exhibit a high degree of anonymity, with fake profile names and no images revealing the owner. The accounts are kept hidden from parents and friends (Whitehead, 2010; Gavin et al., 2008) and serve the purpose of keeping the main account free of pro-ED content. Examples of tweets supporting this research are displayed below.

> **Tweet 1:** Okay so I guess I should tell you guys that [NAME] is not my real name. I use it as a cover in case anyone in real life stumbles across this.

> **Tweet 2:** I have two twitters. One that shows what I actually am, and the other is so other people don't find out.

The tweets are fetched from the pro-ED dataset used in this thesis, and come from two different accounts. As seen in tweet 1, the user clearly states that he/she is using a fake name to hide the fact that he/she has a pro-ED account. The name mentioned in the tweet is censored out in order to protect the anonymity of the user. In tweet 2 the user confirms that he/she has two different accounts, one for pro-ED content and one for non-pro-ED content.

Because of the degree of anonymity and secrecy these accounts usually exhibit, it can be argued that these pro-ED accounts closer represent the personality of the account owner than what the main account does (Yeshua-Katz and Martins, 2013). In a study on pro-ED communities by Gavin et al. (2008), it was discovered that members of pro-ED communities experience acceptance and support from the community and that this allows them to be honest about their disorder. This was also found to be in contrast to the connection they have with their family and offline friends, where they are often afraid to share their thoughts because of the risk of not being understood or accepted.

The fact that pro-ED Twitter accounts are used to share what the account user really feels and thinks, while a main account is kept to make sure the family thinks everything is OK, means that the findings done by Back et al. (2010) might hold for pro-ED Twitter accounts as well. It is, therefore, reasonable to believe that the personality detection done in this thesis is an accurate representation of reality.

5.2. Personality Detection Through Tweets

Personality detection in general is a well-established research field that has been studied for decades. When it comes to automatic detection of personality through tweets, however, the case is a bit different. The existing research on this specific field is sparse, but there are four big contributors that should be mentioned.

The first, and one of the most important contributors in recent years, is the 2013 Workshop and Shared Task on Computational Personality Recognition (Celli et al., 2013). In this workshop, both linguistic and non-linguistic features were applied to analyze personality based on two datasets, one consisting of Facebook status updates and Facebook network information, and one consisting of essays. 16 teams participated in the workshop. All teams used n-grams for the linguistic analysis, but some teams also used categorical features like Part of Speech, and word level features like capital letters and repeated words. Linguistic Inquiry and Word Count (LIWC) based features (described in chapter 4.3) were used by all teams as baselines. Some teams also used other psycholinguistic lexica such as the MRC Psycholinguistic Database (Wilson, 1988) and SentiWordNet

(Baccianella et al., 2010). Other methods used were linguistic nuances and speech act features. Non-linguistic features used by all the teams were Facebook network-properties like size, transitivity, and density. These were also part of the dataset that was provided by the workshop. The most interesting takeaways from this workshop are the features and algorithms that were used to detect personality.

The second big contributor is Kumar et al. (2017). In their research, they looked at how to classify personality through the Big 5 personality traits and Schwartz' values model, which describes human values (Schwartz, 1992). Twitter and Facebook profiles, in combination with essays, were used to predict the personality traits and values while three classifiers were used in experiments; Support Vector Machine trained with Sequential Minimal Optimization (SMO), Simple Logistic Regression (LR), and Random Forest (RF). Before the experiments, the data was pre-processed by stemming and tokenizing it, doing LIWC analysis and normalizing the feature vectors. Which features were significant for which personality trait and value type were pre-analyzed in order to only use the significant features in the final classifier and thereby save computation time and power. N-grams were also added to the LIWC baseline. The performance of the classifier was found to drop by nearly 10% on SVM with uni-grams, while with bi-grams there were no significant changes in performance. Categorical n-grams did, however, get a slightly better performance. Kumar et al. (2017) also added topic models, where they found that 50 topic clusters were most suitable for the task. In pre-processing they removed stop words, but preserved lower and upper case. They found that the best number of clusters were 50 topics, with an average of 19 weighted words. With these weighted topics added to the LIWC baseline, similar results as before were achieved, but the time increased by a factor of 10.

Two other psycholinguistic lexica were also used in addition to the LIWC baseline, namely the Harvard General Inquirer (Stone et al., 1966) and MRC (Wilson, 1988). Sensicon (Tekiroglu et al., 2014), a sensorial lexicon providing a numerical mapping for how much each of the five senses is used to understand a concept, was also included. A final linguistic feature used was speech act features. 11 major categories were used, and 7000 Facebook and Quora[1] utterances were manually annotated and used as input to an SVM based speech act classifier. This classifier used Bag of Words, the presence of *wh* words, the presence of question marks, occurrence of *thanks/thanking* words, POS tags distributions, and sentiment lexica. This gave an improvement of performance of 6.12% in the F_1 score for the Twitter corpus, which was considered a noticeable increase. In addition to the linguistic features, Kumar et al. (2017) also used non-linguistic features based on the social network structure, such as the number of tweets and likes.

The third big contributor is Arnoux et al. (2017) who looked at how to detect personality using 8 times less data than what had previously been used. Their motivation was that a Twitter user has on average only 22 tweets (Burger et al., 2011), so in order to get

[1] `https://www.quora.com/`

a classifier that would work on the average user, they needed to be able to classify personality based on a smaller number of tweets. In order to do this, Gaussian Process (GP) models with word embedding features were used. Words from tweets were extracted and their word embedding representation was averaged into a single vector. Arnoux et al. (2017) also utilized the Twitter 200 dimensional GloVe model (Pennington et al., 2014) and used these vectors as input for the GP model, which was trained for each of the five personality traits. Arnoux et al. collected a ground truth by surveying over 1.3K participants to collect self-reported personality traits as well as tweets. In total, 1323 people with at least 200 non-retweet tweets participated. Most of the participants belonged to the age group 18-24, but participants of all ages were represented. In pre-processing the personality scores were normalized to be in the range of 0-1. The tweets were pre-processed by removing URLs and hashtags, lowercasing the text and removing numbers and punctuation.

As a baseline for comparison, Arnoux et al. (2017) used LIWC with Ridge regression and 3-gram Ridge regression, and the results were compared in three different settings: *Full setting*, *Sampling setting*, and *Real-life setting*. In the full setting, methods were trained and tested using all the tweets of each user. The sampling setting simulated users having a varying number of tweets, so the numbers of tweets included in each user were set to vary. Finally, the real-life setting was trained on a large number of users with a large number of tweets, and tested on a small set of real-life users with a small number of tweets. For the real-life setting, Arnoux et al. collected an additional set of 55 users with on average 28 non-retweet tweets.

Their method achieved a new state-of-the-art performance in the full setting of 33% over the Big 5, which is better than the previous best method. For the sampling setting it was found that LIWC outperforms 3-gram when users have less than 75 tweets. In cases with more than 75 tweets, the 3-gram becomes better. The GloVe model combined with GP was found to perform better overall, being 37% better on 200 tweets and getting better results using only 25 tweets than what the state-of-the-art could get at 200 tweets. Also, in the real-life setting, the GloVe model turned out to be the best with the absolute error being 25% smaller compared to 3-gram and 11% smaller compared to LIWC.

Finally, the fourth and last big contributor is Solomon et al. (2019). They used the Big 5 and Schwartz' sociological behavior model on Twitter data to find out what psycho-sociological facets determine the selection of societal relationships, if these facets can be automatically detected and if they can be used as properties to more accurately detect community structure and predict emerging links. Nearly 600 participants were collected using a web interface, and each participant provided 10 followers and 10 following accounts along with the relationship. Users with less than 100 tweets were discarded, which reduced the number of participants to 559. The participants were from various cultures and ethnic backgrounds. Four classifiers were tested in order to classify personality, values, age, and gender. Solomon et al. (2019) chose to use machine learning algorithms such as SVM,

Simple LR, and RF to develop the models, after inspiration from the WCPR workshop (Celli et al., 2013). For the features, a mix of linguistic and non-linguistic features were used. The linguistic features were various length n-grams, categorical features such as POS and other word-level features like capital letters, repeated words and speech act features. LIWC, MRC and Harvard General Inquirer, and the sensorial lexicon Sensicon (Tekiroglu et al., 2014) were also used. For the non-linguistic features, Solomon et al. used the same as Kumar et al. (2017), which were based on the social network structure. Three machine learning algorithms were used for the personality and Schwartz value classifiers: SVM, LR, and RF, all with ten-fold cross-validation. The SVM algorithm turned out to be the best, outperforming the state-of-the-art system and achieving an average F_1 score of 0.8 for personality detection.

5.3. Personality in Eating Disorder Sufferers

Cervera et al. (2003) conducted a study where they followed a group of young girls, free from eating disorders, in the ages between 12 to 21 for 18 months. It was discovered that people with a high score of neuroticism, in the Big 5 personality model, had a higher risk of developing an eating disorder than people with a lower score. Neuroticism was even found to be more important in the development of eating disorders than low self-esteem.

Multiple research articles (Bollen and Wojciechowski, 2004; Claes et al., 2006) found that people with eating disorders show a higher degree of neuroticism than control groups consisting of people without eating disorders. In a study conducted by Ghaderi and Scott (2000), it was found that openness was significantly higher in people with eating disorders than in healthy people. Another interesting observation with this study is that the researchers found no significant change in the personality of people who developed an eating disorder during the experiment. The personality remained mostly the same as before they got the disease.

A number of studies thus show that neuroticism is present in eating disorder sufferers, meaning that it is possible to make a distinction between the personalities of healthy people and those suffering from eating disorders. This lays grounds for the assumption that looking at personality will provide results that are useful when doing automatic detection of pro-ED accounts.

5.4. Online Pro-ED Communities

A number of studies have been conducted on online pro-ED communities focusing on their content (Borzekowski et al., 2010), the behavior of the members (Whitehead, 2010) and the impact pro-ED sites can have on the viewers (Bardone-Cone and Cass, 2007). Borzekowski et al. (2010) found that 85% of pro-ED websites show thinspiration images or text. They also found that 13% of the sites contained *reverse thinspiration*. Looking at the social networking sites, Facebook and MySpace, Juarascio et al. (2010) found

that the two main reasons why users participate in the pro-ED communities were to get social support and to view and share ED-specific content. Fox et al. (2005) and Whitehead (2010) are both case studies that looked at one website each. For 7 and 6 months respectively, they observed the sites using adapted *face-to-face* ethnographic methods, from Mann and Stewart (2000), to explore the dynamics of the sites. Their findings support the findings of Juarascio et al. (2010). One of the researchers from Fox et al. (2005) were able to join a pro-anorexia website under the impression of being a person with anorexia. This led to a number of conversations and interviews with members of the website. A common factor for the members was that the pro-ana site was a place the members could come to learn how to live with the disease in a safe way. Whitehead (2010) also described the pro-ED community as a form of collective identity for the people in the community.

Several studies have pointed out how *wannarexic* are highly unwanted by the pro-ED community. Boero and Pascoe (2012) published an article that looks at the way pro-ED communities have tackled the problem of wannarexics and how they identify which members are authentic. In their description of the pro-ana anorexic they say:

> The pro-ana anorexic does not seek to hide her body or her disorder, often acts aggressively, actively searches out membership in a pro-ana community, and shows ambivalence about both anorexia and recovery.

They also state that the pro-ED community, as a whole, considers the wannarexic as a threat and therefore tries to expose them. Furthermore, when discovering a wannarexic, the members of the community often act aggressively towards the person. Multiple pro-ED communities focused on trying to keep a small, elite group of true anorexics. This was achieved through group rituals such as check-ins, where the users report weight and diet logs, sharing of pictures and group activities, such as fasts.

One study, looking at the effects of patients diagnosed with an eating disorder viewing pro-ED websites, found that 96% of the participants learned new weight loss/purging methods from the websites and that 69.2% would use these methods after visiting the site (Wilson et al., 2006). Similarly, Csipke and Horne (2007) found that after looking at pro-ED sites, 46% of people who had visited a mental health charity organization would weigh/measure themselves more often. This suggests that the pro-ED sites are indeed harmful to people with eating disorders in the way that they impact the viewers and worsen their condition.

5.5. User Classification on Twitter

This study is based on the findings of Giæver (2018), where five features (unigram, bigram, emoji, biography and Twitter account username) were used together on four machine learning algorithms in order to try to classify pro-ED users on Twitter. The machine learning algorithms that were explored were the Naïve Bayes, logistic regression,

random forest, and Support Vector Machine. The algorithms that produced the highest F_1 score were the SVM and a model consisting of all four machine learning algorithms and a voting classifier. The F_1 score for both algorithms was 0.98. This is the only study done on pro-ED user classification on Twitter.

One of the most important contributions to user classification on Twitter in general is the 2015 *Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality* workshop (Mitchell et al., 2015), which had teams look at the linguistic data from Twitter to detect various mental health problems. Sixteen papers were published in the proceedings of this workshop, looking at mental health problems such as depression, ADHD, Schizophrenia and more.

A participant worth mentioning from this workshop is Coppersmith et al. (2015), which was a shared task where the researchers tried to detect depression and PTSD (Post-Traumatic Stress Disorder) in Twitter users. The researchers attempted to use machine learning classifiers to distinguish the two mental illnesses from each other and from control groups. One of the participants in this shared task was Resnik et al. (2015) who used supervised topic models to detect depression in Twitter users. In their experiments, Resnik et al. used the topic modeling methods LDA and several modifications of it, namely supervised LDA, Supervised Anchor Models, and Supervised Nested LDA. It was found that using the more sophisticated versions of LDA (that used supervision) produced better results than the normal LDA. Resnik et al. also found that by looking at the tweets on a weekly basis instead of looking at all the tweets from each author at once they were able to get better results. This, they pointed out, is intuitive as the mental state changes over time.

Another workshop participant who produced good results was Preoţiuc-Pietro et al. (2015). They used a Bag of Words (BoW) approach, which represented each user as a distribution over words, along with a topic modeling approach. Twitter user metadata such as follower and friend counts, age and gender, and how many tweets the user had posted were also used. Unigrams (n-grams with n=1) as well as several types of word clusters were used as features. The clustering methods used were Brown clusters, a hidden Markov model-based algorithm that clusters words based on what cluster the previous word(s) belongs to, Normalized Pointwise Mutual Information, which uses a similarity matrix between words based on a large reference matrix, Word2Vec, a word embedding approach, GloVe (described in 4.3), LDA (described in 4.1.6) and LDA ER, a different set of LDA topics gathered from Facebook data of an emergency room. Two classifiers were tested: Linear regression, a method that attempts to find a linear relationship between the features and the values to be predicted, and Support Vector Machine (SVM) with a linear kernel. The classifiers were tuned using 10 fold cross-validation. Finally, Preoţiuc-Pietro et al. created a number of classifiers using different feature sets and used a weighted and non-weighted approach to ensemble them into one classifier. The results show that the weighted ensemble classifier was the best for most tasks, giving an average precision of

0.857 across all the tasks and classifiers. Similar results were also found between the SVM and linear regression, but that the SVM was consistently slightly better than the linear regression. When it comes to the features, they all showed similar results to each other, except for the metadata features which had a significantly lower precision than the linguistic features.

6. Data

In order to be able to build the two classification models used in this research, a total of four datasets was acquired. This chapter consists of four parts. The first part describes the four datasets, their origins, composition, and areas of use. All datasets were previously annotated, but one dataset was put through a second round of annotation (a re-annotation). This re-annotation is covered in the second part. All the datasets needed to be pre-processed before they could be utilized, and the third and fourth parts explain the pre-processing steps that each of the datasets underwent.

6.1. Datasets

As mentioned above, a total of four datasets was collected in order to train and test the two classification models developed in this thesis. The first three datasets were used in the construction of the personality detection model, and are described below. These datasets are based on transcriptions from YouTube videos, Twitter posts, and stream-of-consciousness essays, respectively, and all came with annotations of the Big 5 personality traits (see section 2.2 for more information). The last dataset was used to train the pro-ED classification model and is described after the three personality datasets. The pro-ED dataset consists of tweets from a large number of Twitter accounts. Each Twitter account was labeled as either pro-ED, pro-recovery or unrelated based on the content of their associated tweets.

6.1.1. Personality

YouTube transcription dataset
The first dataset used to train the personality detection model is a dataset consisting of transcriptions from the dialog in YouTube videos. The YouTube videos included in this dataset are of the video category vlog. In a vlog the video creator typically films themselves while talking into a camera about topics from their day-to-day life or things that they find interesting at that point in time. The topics can vary from daily thoughts and situations to more complicated themes such as politics or controversies. They are presented in a personal manner, as opposed to being informative and scripted. Vlogs usually take on an informal tone which is similar to the way people write tweets. It is because of this similarity that the dataset was chosen as one of the datasets for the personality model. The dataset contains 404 transcripts in total, each labeled with a personality score following the Big 5 personality format. The personality score consists of five values (one for each of the five personality traits) and has been calculated from a set

of questions (as described in chapter 2.2) answered by the video creators. An example of a personality score is presented below:

VLOG8 5.4 4.8 3.8 4.1 4.2

In the example, the first column represents the vlog ID. This ID is used to link personality score to transcript text which is stored in a separate file. The following columns contain the personality scores in order of extroversion, agreeableness, conscientiousness, emotional stableness (low score indicates neuroticism), and openness. The transcripts themselves consist of only the transcribed text with the filename being the vlog ID. An example taken from part of a transcript is shown below.

> Hi, what's up? Um, today I went to school and um, um, I was late for my first class because, um, it was really hard for me to get up 'cause it's a Friday. Thank God it's Friday!

MyPersonality Twitter dataset
The second dataset is a Twitter dataset collected by the MyPersonality Facebook application.[1] This application allowed users to take a personality test and receive a Big 5 personality score based on their test answers. It also made it possible for the users to provide their Twitter userID if they owned a Twitter account. The Twitter dataset was created by collecting the tweets from the users who had provided their Twitter userID and store them along with a value in the range 1.00-5.00, for each of the five personality traits, to describe their personality scores. This dataset contains a total of 8947 tweets from 172 users, making it a small dataset, but considering that the data is very similar to the pro-ED dataset the classifier will be applied to, it was still deemed highly useful. To protect the privacy of the users, the dataset did not provide the Twitter userID, but instead replaced it by a unique, anonymous ID to distinguish which tweets belong to which user. Other metrics were also included in the dataset, but only the ID, tweets and personality scores were extracted from the dataset and used in the work presented in this thesis.

MyPersonality Essays dataset
The third dataset is an essay dataset provided by the same MyPersonality app as the Twitter dataset mentioned above. This dataset contained stream-of-consciousness essays that were written by a total of 2467 psychology students. A stream-of-consciousness essay is an essay where the author writes whatever comes up in their mind, and is therefore very informal. This makes it a relevant type of data to use for a classifier that will be used for tweets, as the linguistic form is often very similar to that of tweets. Below is an example of the first few sentences of one essay from the dataset.

> Well here we go with the stream of consciousness essay. I used to do things like this in high school sometimes. They were pretty interesting but I often

[1] `https://sites.google.com/michalkosinski.com/mypersonality`

> find myself with a lack of things to say. I normally consider myself someone who gets straight to the point. I wonder if I should hit enter any time to send this back to the front. Maybe I'll fix it later. My friend is playing guitar in my room now. Sort of playing anyway. More like messing with it. He's still learning.

In addition to the essay texts, this dataset also contained binary values for each of the personality traits in the Big 5 model. This means that for each of the personality traits, the dataset only contained information about whether the author was considered to have the trait or not, as opposed to the floating-point regression values that the other two datasets have.

6.1.2. Pro-Eating Disorder

The fourth and last of the datasets is a Twitter dataset used to create the pro-eating disorder classification model. Giæver (2018) created this dataset as part of her Master's Thesis by downloading tweets from Twitter accounts that met a set of search criteria. The dataset consists of 7096 Twitter accounts and 10.7 million tweets. The dataset has been manually annotated by Giæver with labels for pro-ED, pro-recovery and accounts without any eating disorder related content, labeled unrelated accounts. Of the 7096 users, 33% were pro-ED, 11% pro-recovery, and 56% unrelated to eating disorders.

Giæver (2018) provided two different versions of the dataset. The first version consisted of a Twitter userID and the associated annotation label. In order to use this dataset, it was necessary to download all the tweets belonging to each Twitter userID. This was the desired option as this would make it possible to build a raw dataset free of any possible modifications that might follow a pre-built dataset. As it turned out that a large portion of the tweets was deleted or made unavailable by the time this thesis was written, using this option became impossible. The second version that Giæver provided consisted of the Twitter userIDs, their annotation label as well as all their associated tweets. This version of the dataset is unavailable to the public, but was obtained through internal distribution as a result of the authors being in the same research group as Giæver. In this version of the dataset, the tweets had already been put through to some modifications, but in order to avoid loosing too much data, it was decided to use this version of the dataset. The modifications done to the tweets consisted of the removal of non-English tweets, removal of symbols, replacing task-relevant entities such as URLs and mentions with placeholders, and replacing emojis with a descriptive term for what the emoji signals. Some accounts, but not all, also had all numbers in their tweets removed.

6.2. Annotations

The three personality datasets were labeled based on answers to questionnaires that the participants had answered. Because of this, it was impossible for the authors to do any further annotation on these datasets. The nature of how the personality labels

were created also meant that it was not necessary to annotate the datasets further. The pro-ED dataset, however, was in need of a second round of annotation.

Each account in the Pro-ED dataset was labeled with one of three annotation labels: pro_ed, pro_recovery or unrelated. The labeling was a result of a manual annotation process done by Giæver (2018) and three control annotators. Giæver annotated the entire dataset by herself and used the three control annotators to check portions of the dataset to make sure that the annotations were appropriate. Each control annotator was tasked with annotating 100 accounts, where each account contained a subset of 200 tweets. Out of the 100 accounts, the three control annotators had a total of 30 overlapping accounts, which were used to measure the agreement between the control annotators as well as the agreement with Giæver. The agreement measurements showed that there were few deviations in the annotation done by the control annotators and Giæver, and as a result the agreement was close to perfect.

Even though the agreement between the annotators was high, only a small portion of the dataset was checked by control annotators, and an even smaller portion lay the grounds for the agreement measurement. This means that most of the dataset has been annotated by only one person and there might, therefore, be traces of mistakes and subjective choices in the dataset. As a result, it was deemed beneficial to do a second round of annotation.

6.2.1. Categorizing Pro-ED Accounts

Before conducting a re-annotation of the pro-ED dataset, it was important to make sure that the method and reasoning behind the annotation done by Giæver (2018) were fully understood. It was also important to create a common understanding between the authors of how to categorize an account as either a pro-ED account, a pro-recovery account or an unrelated account as it would be this common understanding that would make sure that the annotation done by the two authors would be of the same quality.

Giæver defined a pro-ED account as an account where a positive pro-ED attitude was displayed at least once, either in the tweets, retweets or profile information. This definition lay the ground for four inclusion criteria in which an account was considered a pro-ED account if the tweets, retweets or profile information:

(1) Included a self-identification as pro-ED, or

(2) Expressed a desire for emaciation, or

(3) Ascribed to a pro-ED event, or

(4) Encouraged extreme weight control methods

The three first criteria were fetched from Arseniev-Koehler et al. (2016) while the fourth was a contribution of her own. Satisfying one of the criteria was considered a display of

Table 6.1.: Example Tweets and Pro-ED Inclusion Criteria Satisfaction

Examples:	**Inclusion criteria sattisfied:**	**Label:**
Pretty girls don't eat #skinny #bones #bonespo #promia	1, 2, 4	Pro-ED
Don't pretend curves are sexy. Bones are sexy	2	Pro-ED
I'm not pro anorexia, Im just proana and proed	1	Pro-ED
I wanna see how long I can go without eating before I finally give in	4	Pro-ED
#edproblems #ana #thinspo #skinny4xmas	3	Pro-ED
I will vomit for love #skinny4xmas #skinny #bulimia #edproblems	1, 3, 4	Pro-ED

a positive pro-ED attitude and the account would be labeled as pro-ED. Based on the information gathered about the pro-ED community and eating disorders (see chapter 2.1) it was decided that the four evaluation criteria would be used as is to categorize pro-ED accounts in the re-annotation.

If an account did not fulfill any of the pro-ED inclusion criteria, it would be labeled as either pro-recovery or unrelated. For an account to be pro-recovery it was decided that the account had to express recovery focused content in relation to eating disorders. The focus on recovery in relation to eating disorders was important, as many Twitter accounts contain recovery related content but with a focus on other illnesses such as cancer, drug addiction, surgery or mental illnesses in general, rather than on EDs. Labeling these accounts as pro-recovery would introduce unwanted noise to the pro-ED classification model. As a result, accounts related to recovery-oriented themes, other than eating disorders, would therefore not be labeled as pro-recovery. This is a bit different from the way Giæver labeled pro-recovery accounts and the re-annotation might, therefore, affect the category distribution in the dataset.

For an account to be labeled as unrelated, it had to be an account that would fail to fulfill any of the inclusion criteria for pro-ED, as well as not contain any recovery focused content in relation to eating disorders. In short, all the accounts not fitting into either being labeled as pro-ED or pro-recovery would be labeled as unrelated. This way of defining unrelated accounts was in agreement with the way Giæver defined unrelated accounts.

An example of how tweets can fulfill the inclusion criteria described above can be seen in table 6.1. The table contains a series of example tweets and each tweet is linked to one or multiple inclusion criteria that it fulfills. The tweets are also marked with a label based on the inclusion criteria. All the tweets belong to the pro-ED category as a result of

Table 6.2.: Example Tweets for Pro-Recovery and Unrelated Categories

Examples:	**Inclusion criteria sattisfied:**	**Label:**
What eating disorder survivors want you to know about recovery #edrecovery	-	Pro-recovery
I ate my entire meal today and I am so proud	-	Pro-recovery
Face yourself honestly on a daily basis #edwarrior #edrecovery #onedayatatime	-	Pro-recovery
Trust yourself no matter what anyone else thinks	-	Unrelated
New devestating swedish poll, it shows that 1 in 5 women are afraid of being sexually assaulted	-	Unrelated
Second round of treatment today, wish me luck #recovery #cancer	-	Unrelated

fulfilling one or multiple of the inclusion criteria. Table 6.2, on the other hand, contains a series of example tweets that do not fulfill any inclusion criteria and have therefore got the category labels of either pro-recovery or unrelated.

6.2.2. Re-Annotation

Having established a common understanding of how to categorize a Twitter account, it was time to begin the re-annotation of the pro-ED dataset. The dataset was divided into two equal parts consisting of 3548 accounts each. For each account, the username, biography and a subset of 200 randomly selected tweets were extracted. It was decided that 200 tweets would provide enough information about the account to be able to label it, while at the same time limit the amount of data enough to make sure that the annotation process was done effectively. It was also decided that the tweets to include in this subset would be selected at random from the user, as opposed to choosing 200 consecutive tweets. This was done because a user might tweet more about pro-ED in periods, and by choosing tweets at random it would give a broader perspective of the account throughout time. In order to make sure that it would be possible to measure the quality and agreement of the annotation, the two parts of the dataset were expanded with 500 accounts each, resulting in an overlap of 1000 accounts. The two authors annotated one part of the dataset each, meaning that the 1000 overlapping accounts were annotated by both authors in addition to Giæver (2018). The remainder of the dataset was annotated by one of the authors as well as Giæver.

The category distribution in the dataset prior to re-annotation is described in table 6.3. Unrelated is the largest of the three categories, with pro-ED being the second largest, and pro-recovery the smallest. This is also reflected in the number of tweets belonging to each category. Unrelated contains 63.4% of the tweets in the dataset, pro-ED contains

Table 6.3.: Composition of Pro-ED Dataset Prior to Re-Annotation

Label	Number of users	Number of tweets
Pro-Eating Disorder	2 355	2 618 120
Pro-Recovery	804	1 305 670
Unrelated	3 937	6 821 083
Total	**7 096**	**10 744 873**

Table 6.4.: Composition of Pro-ED Dataset After Re-Annotation

Label	Number of users	Number of tweets
Pro-Eating Disorder	2 361	2 625 081
Pro-Recovery	693	1 054 137
Unrelated	4 042	7 065 655
Total	**7 096**	**10 744 873**

24.4% of the tweets and pro-recovery contains 12.2% of the tweets.

Table 6.4 displays the category distribution of the dataset after re-annotation. Unrelated has remained the largest category, pro-ED the second largest and pro-recovery the smallest. Due to the different policy between Giæver (2018) and the authors in what defines a pro-recovery account, the amount of pro-recovery accounts has decreased by 111 accounts and unrelated accounts has increased by 105 accounts. The number of pro-ED accounts has increased by 6. The dataset now consists of 65.8% of tweets labeled unrelated, 24.4% with the pro-ED label and 9.8% pro-recovery. Out of the three categories, only pro-ED contained the same percentage of tweets as before the re-annotation, while unrelated increased, and pro-recovery decreased by 2.4% each.

6.2.3. Inter Annotator Agreement

As mentioned above, 1000 accounts were made to overlap when annotating to make sure that the quality and agreement of the annotation could be measured. These 1000 accounts were annotated by both authors and by Giæver (2018), meaning that it would be possible to measure not only the agreement between the authors but also the agreement between the authors and Giæver. As mentioned in chapter 4.2.2, two methods exist to measure agreement: Cohen's Kappa and Fleiss' Kappa.

Cohen's Kappa was used to measure the agreement between the two authors. 1000 accounts were used as the basis for the calculation. The agreement was calculated in two different ways. The first way as a multi-class evaluation with all three categories (pro-ED, pro-recovery and unrelated), the second way as a binary evaluation (pro-ED and not pro-ED). Pro-recovery and unrelated accounts were combined into a not pro-ED class. As seen in table 6.5, both the multi-class and the binary measurement display a high

Table 6.5.: Cohen's Kappa Agreement Between Authors

	Multi-class	**binary**
Authors	0.98	0.97

Table 6.6.: Fleiss' Kappa Agreement Between Authors and Giæver

	Multi-class	**binary**
Authors + Giæver	0.96	0.99

degree of agreement between the annotation done by the two authors. There is a slightly higher agreement in the multi-class measurement. The high degree of agreement point to an annotation done according to the inclusion criteria, and that the annotation was done with few errors and subjective choices.

Fleiss' Kappa was used to measure the agreement between the two authors and Giæver. The measurement was done both as multi-class and as binary measurement. Table 6.6 contains the results of the measurements. As can be seen, both multi-class and binary measurement show nearly perfect agreement between all the annotators, with binary having the highest score of 0.99. A possible reason for the difference in the multi-class and binary agreement measurement might be that the two authors had a different view on what made an account a pro-recovery account than what Giæver had. Some accounts were therefore changed from being classified as pro-recovery to being classified as unrelated. This lead to some disagreement between the authors and Giæver in the annotations, which again affects the multi-class result. In the binary evaluation, pro-recovery and unrelated were merged into one class, which lead to the high agreement result.

6.3. Personality Pre-Processing

None of the three datasets used in the personality categorization model contained raw data. They had all been modified to some degree at the time they were obtained by the authors. In order to be able to use the three datasets together as one large dataset, it was necessary to put them through a series of pre-processing steps. The main focus of the pre-processing steps was to make all three datasets as similar in structure and composition to the pro-ED dataset as possible. This was because the regression model trained on the personality datasets would be used on the pro-ED dataset in the final classifier. If the personality datasets were different from the pro-ED dataset it would be difficult for the categorization model to categorize the pro-ED dataset accurately. Since the datasets were collected from different media and had different types of values for the annotation labels, different pre-processing steps were required for each of the datasets. The pre-processing done to the YouTube transcription dataset is explained first followed by the pre-processing steps done to the MyPersonality Twitter dataset and the MyPersonality Essays dataset.

Table 6.7.: Personality Dataset Pre-Processing

Before	After
, . ? - "	
um, uh, ah,	
I'm	im
ok\n	ok

6.3.1. YouTube Transcription Dataset

In order to get this dataset as similar to the pro-ED dataset as possible, a series of pre-processing steps focusing on text processing were applied to the YouTube transcripts. As seen in table 6.7, four pre-processing steps were applied. The transcripts contained numerous double dashes (–), quotation marks ("), and other symbols such as apostrophes and commas. Seeing as all these symbols had been removed from the pro-ED dataset, it was decided that they should be removed from this dataset as well. As a result of the transcripts being created from informal, oral speech, the words *um*, *uh* and *ah* appears a lot. These words are generally used in speech as discourse markers, but are rarely used in written text, including tweets. Because of this, all occurrences of these words were deleted from the dataset. All text was set to lowercase, as this was something that had been done to the pro-ED dataset as well. Finally, there were a large number of newline marks. Because these did not provide any content related information they were removed as well.

An example of how a transcript from the dataset looks was shown in chapter 6.1.1. When looking at the same example after the pre-processing there is a noticeable difference.

> hi whats up today i went to school and i was late for my first class because it was really hard for me to get up cause its a friday thank god its friday

The text now seems more coherent and resembles a tweet much more than what it did before the pre-processing.

6.3.2. MyPersonality Twitter Dataset

Out of the three datasets, this dataset was the one with the fewest modifications from being a raw dataset. This meant that, in addition to the steps described in table 6.7, this dataset needed a bit more pre-processing than the others. This was done in order to make it as similar to the pro-ED dataset as possible. Both the YouTube transcription and the MyPersonality Essays datasets contained long coherent text strings. This Twitter dataset contained all the tweets from a single account as separate tweets and not as one long coherent text. In order to make this dataset similar to the other datasets, all the tweets from a single account were concatenated into one long string. The data was then

checked for non-English language. Because the pro-ED dataset only contains English tweets, all tweets that were written in languages other than English were removed from the dataset. The next step focused on replacing URLs and mentions, with respectively *URL* and *MENTION*, in all the tweets, making all text lowercase, and removing symbols and punctuation. An example of how a tweet looked before the pre-processing and after pre-processing can be seen below:

Before:

> ATTENTION EVERYONE!!! Vote for the short ""AFTER HOURS"" to support *PROPNAME* - our local Michigan talent and a wonderful person. Repost please!!! http:////www.thirteen.org//sites//reel13//category//vote//

After:

> attention everyone vote for the short after hours to support MENTION our local michigan talent and a wonderful person repost please URL

The pre-processing steps left the dataset with 169 out of 172 Twitter accounts, meaning that only three accounts were deleted.

6.3.3. MyPersonality Essays Dataset

The essays in this dataset were pre-processed in the same manner as the YouTube and Twitter datasets. However, the personality labels indicating the personality score for the essays came in the form of *yes* or *no* values. This differed from the way the other two datasets were labeled. In order to get consistent labeling across the datasets, two different approaches were tested for converting the binary classification labels into numerical values. These tests are elaborated on in chapter 8.1.1. The experiments ended with the labeling going from looking like the first values to the second values shown below:

Before:

> [ID, TEXT, y, y, n, n, y]

After:

> [ID, TEXT, 3.76, 4.2, 1.35, 2.1, 3.5]

6.4. Pro-Eating Disorder Pre-Processing

As mentioned in chapter 6.1.2, the pro-ED dataset had already been modified to some extent by Giæver (2018). While some of the experiments in this thesis were done with

the dataset as it was when provided, multiple pre-processing steps were applied to the dataset in order to make the data optimized for the final classifiers. An analysis of the dataset was conducted in order to be able to detect the pre-processing steps needed to optimize the dataset. This analysis is presented first. The pre-processing steps done in order to optimize the dataset, as well as the reasoning behind each step, are then presented, followed by the results of the pre-processing in terms of how it affected the size, structure, and composition of the dataset.

6.4.1. Dataset analysis

The annotation of the pro-ED dataset provided a lot of insight into the structure and composition of the dataset. It also revealed a large number of problem areas which had arisen as a result of the modifications done by Giæver. It turned out that a lot of symbols, special characters, and foreign languages had not been properly processed, leading to partly processed byte strings in the tweet texts. Another issue that was discovered was that there were some accounts that had all numbers removed from the tweets while other accounts had not. This created inconsistency in the dataset, which could be unfortunate for the quality of the experiments. Furthermore, all symbols had been removed from the tweet texts, but the way they had been removed varied between either being replaced by a space character or by simply being deleted. Words such as *don't* were therefore represented as either *don t* or *dont* in the initial dataset. This meant that a word that had the same meaning was represented as two different words, which again is a possible source of error. Another similar issue was that the symbol *&* was represented as either *amp* or *and*. It was decided that a consistent dataset would be the most beneficial and as such the issues discovered during the analysis lay the ground for the following pre-processing steps.

6.4.2. Pre-processing steps

The pre-processing process followed these eight steps:

1 Decode non-ASCII characters

2 Delete remaining encoded strings

3 Merge single letters with associated words

4 Replace *amp* with *and*

5 Delete numbers

6 Delete non-English language tweets

7 Combine all tweets per account into one string

8 Delete accounts with less than 20 remaining tweets

Decode non-ASCII characters
When modifying the dataset, Giæver had encoded the strings in such a way that she encoded all non-ASCII characters as byte strings. This would normally be fine, but considering she had later removed all symbols, including the backslash which is used in the byte strings and sometimes numbers, decoding the text was no longer such an easy task. An example of how a byte string had been modified in the dataset, how it was supposed to look like, and how the decoded symbol looks like can be seen below:

xedx95xa0xeaxbaxbc

\xed\x95\xa0\xea\xba\xbc

할꺼

The example byte string was taken from a tweet written in Korean and as such the Korean symbols had not been decoded. The way this issue was solved is described in chapter 8.2.1.

Delete remaining encoded strings
Most of the byte strings were decoded in the process mentioned above, but in some cases, the byte strings were impossible to decode due to numbers having been removed from some of the Twitter accounts. Without the numbers, the byte strings were left incomplete and it was no possible way of knowing what the original meaning of the bytes was. An example of a byte string that had been modified and could not be decoded can be seen below:

xe x xafxe x xbfxe x x fxe

Having to delete these meant a possible loss of data, but after seeing that many of the successfully decoded byte strings proved to be either symbols or non-English language, which were both to be deleted at a later stage in the pre-processing, it likely did not take away too much significant data. The deletion would also remove unwanted noise from the dataset, which could justify a possible data loss. With this in mind, it was decided that the best option would be to delete the remaining byte strings from the dataset. An experiment was done in order to make sure that the deletion would not harm the dataset. The way the deletion was done, and the result of the experiment, is described in chapter 8.2.1.

Merge single letters with associated words
As a result of the modifications done by Giæver (2018) some words in the dataset were represented in two different ways. One of these words were *don't*, which could be represented as either *don t* or *dont*. This meant that the same word would be counted as two different words when present in a feature. To prevent this, it was decided that all standalone occurrences of the characters *s*, *m*, and *t* would be merged with the previous

word. *Don t* would then become *dont*, *can t* would be *cant* and *i m* would be *im*. As with all the pre-processing steps mentioned above, an experiment was carried out in order to make sure the step was beneficial and not harmful to the dataset. The way this pre-processing step was done, and the results of doing it, is described in chapter 8.2.1.

Replace *amp* with *and*
In a similar manner to the standalone characters, the symbol & was represented as just *amp* in the dataset. The reason for this might be that, at the time of tweet download, the symbol might have been in the HTML format, *&amp*, and when Giæver modified the dataset, the & symbol was removed together with the removal of all other symbols. A lot of tweets had used the word *and* to represent &, meaning that both *and* and *amp* was used to represent the same thing. An experiment was done to see if it would be beneficial to replace *amp* with *and* (see chapter 8.2.1). Replacing *amp* with *and* would ensure that words with the exact same meaning would be represented in the same way. It would also assure that it would be properly handled by the Natural Language Processing tools such as in the removal of stop words or in Part of Speech tagging.

Delete numbers
Since many of the accounts in the dataset already had their numbers removed, it was decided to strive for consistency and delete the remaining numbers in the dataset. The number deletion did not only delete numbers present in tweets, but also entire tweets consisting of only numbers. An experiment was done in order to make sure that the deletion would not harm the potential of the dataset in any way. This is further explained in chapter 8.2.1. The deletion of numbers resulted in the deletion of 39 368 tweets.

Delete non-English language tweets
Even though the dataset was mainly in English, some accounts and tweets were in foreign languages. This became even clearer after having decoded the byte strings. As the machine learning classification model created in this thesis focuses only on English text, all tweets that were detected as other languages were removed from the dataset. Giæver (2018) explained in her thesis that she had also deleted all non-English tweets, meaning that this pre-processing step was just an extension of her work. After the non-English tweets were removed, only one account was deleted and 9 824 116 tweets remained, 881 389 less than after the number deletion.

Combine all tweets per account into one string
Instead of keeping each tweet as a separate row in the dataset, the dataset was restructured by adding all tweets belonging to an account into one continuous string. This was done because the classifier was supposed to work on Twitter account level and not on single tweets. It was also done to decrease the size of the dataset to make the algorithm more efficient. By adding all the tweets into one long string, the text would resemble a document instead of a short line of text. This fits the format of the datasets used to train the personality categorization model.

Table 6.8.: Composition of Pro-ED Dataset After Re-Annotation and Pre-Processing

Label	Number of users	Number of tweets
Pro-Eating Disorder	2 293	2 384 068
Pro-Recovery	675	1 019 780
Unrelated	3 856	6 417 259
Total	**6 824**	**9 821 107**

Delete Accounts with Less than 20 Remaining Tweets
Giæver mentions in her thesis that accounts consisting of fewer than 20 tweets were deleted. However, some accounts seem to have escaped this deletion and others might have arisen due to the deletion of non-English tweets and numbers. As a result, it was decided that the remaining accounts with less than 20 tweets would be removed. In total, 271 accounts were deleted and the total number of tweets left in the dataset was 9 821 107. This is a further reduction of 3 009 tweets from after removing non-English tweets.

6.4.3. Dataset after Pre-processing

The pre-processing resulted in a new and more consistent dataset with noticeable differences to the dataset received from Giæver (2018). Table 6.8 displays the composition and structure of the new dataset. The total amount of tweets diminished by 923 766 compared to the initial dataset depicted in table 6.3. One reason for this is the removal of numbers in tweets, which resulted in the deletion of tweets consisting of only numbers. Another reason is that a lot of the encoded tweets turned out to be non-English tweets or symbols which were deleted in the language check. 272 accounts were deleted as a result of the removal of non-English tweets and the removal of accounts with less than 20 tweets. Out of the 272 deleted accounts, 68 were labeled as pro-ED, 18 as pro-recovery and 186 as unrelated. This meant that all three label groups were reduced compared to how the dataset looked after the re-annotation (table 6.4). The final dataset consists of 65.3% tweets labeled unrelated, 24.3% labeled pro-ED and 10.4% labeled pro-recovery. This is a slight decrease in unrelated and pro-ED labeled tweets, and an increase in pro-recovery.

In addition to pre-processing the tweets themselves, the rest of the dataset (username, bio, location) were also pre-processed using pre-processing step 1 to 5 (described in chapter 6.4.2). This was done after the pre-processing experiments that decided which steps would be included. The only data element not objected to pre-processing was the Twitter userID. The userID is quite restricted in terms of what the user can choose to put there, and therefore it was not considered necessary to pre-process.

7. Architecture

One thing that stood out as a common theme in the related literature was that there are four general steps to building a text classification model. These four steps are; data collection, pre-processing, feature extraction and model building. The previous chapter described the first two steps (data collection and pre-processing). This chapter continues with the two remaining steps (feature extraction and model building). The features that were chosen to be used in experiments, and the process of extracting them, are explained in the first part of this chapter. The machine learning algorithms used in the personality and pro-ED classification experiments are described in the second part. Finally, the third part covers the architecture and construction of the machine learning classification models that were created in this thesis.

7.1. Feature Extraction

Feature extraction is an important part of creating a machine learning model, and the choice of features has a huge impact on the performance of the model. The four datasets used in this thesis were analyzed in order to find which feature groups would be beneficial to the classification. Often, a large number of features is optimal for creating good classifiers, but this can also make the classifier slow and take up a lot of computational power. Having too many features can also make the classifier overfit to the training data, leading to worse performance on test data. To handle this problem, the various features that were considered were all tested one by one, and with varying parameters to find the most relevant features and the most optimal configurations of these features. The feature groups can be divided into two groups: the features used by Giæver (2018) in her thesis and the new features explored in this thesis. The tools used for feature extraction is described in chapter 4.3

7.1.1. Feature Groups Used by Giæver

Giæver (2018) used five different feature groups in her experiments. These feature groups were chosen based on an extensive data analysis of the pro-ED dataset that she carried out as part of her thesis. The analysis showed that there were noticeable differences in the use of words based on whether the account was labeled as pro-ED, unrelated or pro-recovery. Different emojis were also used in addition to the words used in account display names and account usernames. Many pro-ED account display names and account usernames contained pro-ED related words, which could be helpful in the automatic pro-ED account detection. The result of her analysis showed that it was possible to

differentiate between the three types of accounts based on words and emojis. Because the goal of this thesis was to improve upon the results made by Giæver, the features used in her thesis were also tested in the experiments in this thesis.

Giæver used SciKit-learn to extract her features, which means that in order to be able to recreate her results and possibly improve upon them, it was necessary to use the same procedure in this thesis as well. In order to convert the long strings of tweet text, the module feature_extraction.text was used. This module contains the TfidfVectorizer class, which implements tokenization, occurrence counting, and Term Frequency-Inverse Document Frequency (TF-IDF) weighting. With SciKit-learn it is possible to specify a number of parameters, such as the value of n in n-grams extraction or a lower or upper boundary for the vocabulary size. In this case, unless otherwise specified, default parameters were used. The five feature groups used by Giæver (2018) are listed below:

- **Unigram features from tweets**
  This feature group consists of single words extracted from the tweet text of each account. This feature was extracted by using the TfidfVectorizer with ngram_range parameters of (1,1).
- **Bigram features from tweets**
  The bigram feature group consists of two adjacent words and was extracted by changing the ngram_range parameters of the TfidfVectorizer to (2,2).
- **Emoji features from tweets**
  Emoji features were extracted by running only the emoji placeholder tags found in tweets through the TfidfVectorizer as unigrams.
- **Biography features**
  This feature group consists of unigrams and bigrams made from the biographies of the Twitter accounts. This was achieved by setting the ngram_range parameters to (1,2).
- **Username features**
  Username features were extracted from account display names and usernames by using n-grams on character level from length 3 up to 15 (maximum length of usernames). This was done by setting the analyzer parameters of the TfidfVectorizer to 'char' instead of 'word' and the ngram_range parameter to (3, 15).

In addition to being used to recreate the experiment from Giæver (2018), the features were also tested in combination with the new features described in the next section.

7.1.2. New Feature Groups

To get the best possible classifier, with the highest possible performance, five new features were extracted and different combinations of these were tested. The five new features tested in this thesis are listed below.

- **Topic Modeling from tweets**
  Topic modeling was applied in the form of Latent Dirichlet Allocation (LDA). The motivation for using these features was the work published by Resnik et al. (2015), where it was found that these features provided great performance when classifying depression in tweets (see chapter 5.5 for more information). The experiments used the gensim[1] *LdaMulticore* model with 10 topics and a dictionary of 10000 words. Words that appeared in less than 15 users and words that appeared in more than 50% of the users were excluded.

- **Word Embedding from tweets**
  The word embedding method Global Vectors (GloVe) was used due to the good results achieved with this method by Arnoux et al. (2017). It is possible to train a GloVe model manually, and this can be a good choice because the model can then be trained on the same data it will be used on. In this case, however, it was decided to download a pre-trained model[2] which had been trained on 2B Tweets. The reason for choosing a pre-trained model was that because it had been trained on such a large number of tweets it was likely to work well with the datasets used in this thesis. This feature was tested both for the personality detection algorithm, and the pro-ED classifier.

- **N-grams from tweets**
  N-grams on word level were extracted using TfidfVectorizer from the SciKit-learn library, with ngram_range parameters set to (1,3) and stopwords removed. The features were extracted as described in the previous section. The reason for using n-grams as a feature was because it often makes sense to look at words in combination instead of on a one-by-one basis. N-grams had already been tested by Giæver (2018), but due to the changes to the dataset in pre-processing, it was decided to test it for the experiments in this thesis as well.

- **Part of Speech from tweets**
  Part of Speech (POS) tags were extracted using ARK tweet NLP (see chapter 4.3). This POS tagger was chosen because it is designed specifically for Twitter data and is, therefore, able to understand many of the abbreviations and linguistic styles that are present in tweets. Using POS tags as a feature is interesting because they can capture writing style and highlight things such as how often the account tweets contain word groups like pronouns and adjectives. It could be worth testing if these things make a difference both in the personality detection model and in the pro-ED classifier. Giæver (2018) reported in her research that using POS yielded unsatisfying results and was therefore excluded from further experiments, but because of the use of a different tagger that is created for tweets, as well as the different annotation and pre-processing done in this thesis, it was decided that it was worth testing if POS tags could be a useful feature. The POS tags

[1] `https://radimrehurek.com/gensim/index.html`
[2] `https://nlp.stanford.edu/projects/glove/`

were, however, only used as a feature for the pro-ED classifier and not with the personality detection. This was because collecting the POS tags required that the library opened and closed Java processes for each tweet in order to access the underlying Java program. This turned out to be a very computationally hard process, and there was not enough time to extract the tags from all datasets.

- **LIWC**
  Linguistic Inquiry and Word Count (LIWC) analysis has been utilized in many studies with similar goals as this thesis, often to establish a baseline. For example, in the 2013 Workshop and Shared Task on Computational Personality Recognition (Celli et al., 2013) all teams used LIWC as their baseline, as did Kumar et al. (2017) and Arnoux et al. (2017) (see chapter 5.2 for more information). Many of the categories that are provided in the LIWC analysis, such as affective processes, social processes, and cognitive processes, could be useful in classifying both personality and pro-ED accounts. It is reasonable to assume that these types of categories would make a good feature for both the personality detection model and the pro-ED classifier.

7.2. Implementation of Machine Learning Algorithms

A total of five machine learning algorithms were used in this thesis, all of which are described in chapter 3.2. While the first two of these (Support Vector Machine and Gaussian Process) came from related research and have been proven to produce results of good quality (see chapter 5.2), the three remaining algorithms (K-Nearest Neighbors, Ridge Regression, and Multilayer Perceptron) were chosen by the authors. The reason for choosing these algorithms was that they were thought to be good representations of different types of machine learning algorithms.

7.2.1. Support Vector Machine

As described in chapter 3.2.1, Support Vector Machine (SVM) is a popular machine learning algorithm used for both regression and classification tasks. SVMs have been shown to provide good results in much of the related research on both personality (Kumar et al., 2017; Solomon et al., 2019), and for detecting pro-ED accounts in Twitter (Giæver, 2018). In addition to producing good results in the related research, SVMs are often relatively efficient compared to other methods due to the kernel computations. For these reasons, it seemed like a good algorithm to include in the experiments for this thesis. As mentioned before, the SVM algorithm was the one that produced the best results for Giæver (2018). Because these results were chosen as a foundation for baseline creation, it was necessary to be able to recreate the results she got using the SVM algorithm. The algorithm runs Sequential Minimal Optimization in the background in order to speed up the calculations.

In all experiments on features for the pro-ED classifier, SVM was used with a linear kernel. This was chosen because it makes it possible to use the built-in cross_-val_score from the sklearn.model_selection. This allows for cross-validated scores for the precision, recall, and F_1 score metrics without implementing the cross-validation manually. Having this ability makes it a good choice when comparing the performance of a feature. Linear SVM is also known to be fast, which is a good quality when many experiments are to be conducted.

7.2.2. Gaussian Process

Gaussian Process (GP) was chosen as a result of the study done by Arnoux et al. (2017). This study represented a new state-of-the-art method of automatic personality detection, which used less data than previous state-of-the-art methods (see chapter 5.2 for details). It was decided it would be interesting to see how the results from both the personality detection and the pro-ED classification models when using GP would compare to the results obtained from the other algorithms.

GP was implemented using the GaussianProcessClassifier and GaussianProcessRegressor from the sklearn.gaussian_process package. In some experiments, the model was tested with different kernels, but unless otherwise specified, the kernel used was a sum of dot-product and white-kernel, both with all default parameters. The dot-product kernel was used because, being obtained by linear regression, it is a simple and fast kernel. The white-kernel was chosen because it can be used to explain the noise in the data. This was considered to be particularly important for the personality dataset, which it is reasonable to assume included some noise since the personality scores were self-reported.

7.2.3. K-Nearest Neighbors

The K-Nearest Neighbors (k-NN) algorithm was chosen because of the algorithm's ability to classify elements based on the surrounding elements, as mentioned in chapter 3.2.3. When used for classification, the algorithm was implemented by using KNeighborsClassifier and KNeighborsRegressor from the package sklearn.neighbors. In many cases, using the distance metric *Hamming* is preferable for text classification (as described in chapter 3.2.3). However, using this metric requires integer input, which was not what was produced by many of the features chosen. For this reason, the metric parameter was left at the default value, *Minkowski*.

7.2.4. Ridge Regression

Ridge Regression (RR) was included in the experiments because it is known to be very good at handling text classification problems. Because of the large input space of these types of problems, more complex models will often overfit, leading to linear models such as RR being better at generalizing to new data. In the experiments with the personality dataset, the algorithm was implemented using sklearn.linear_model.Ridge, using all

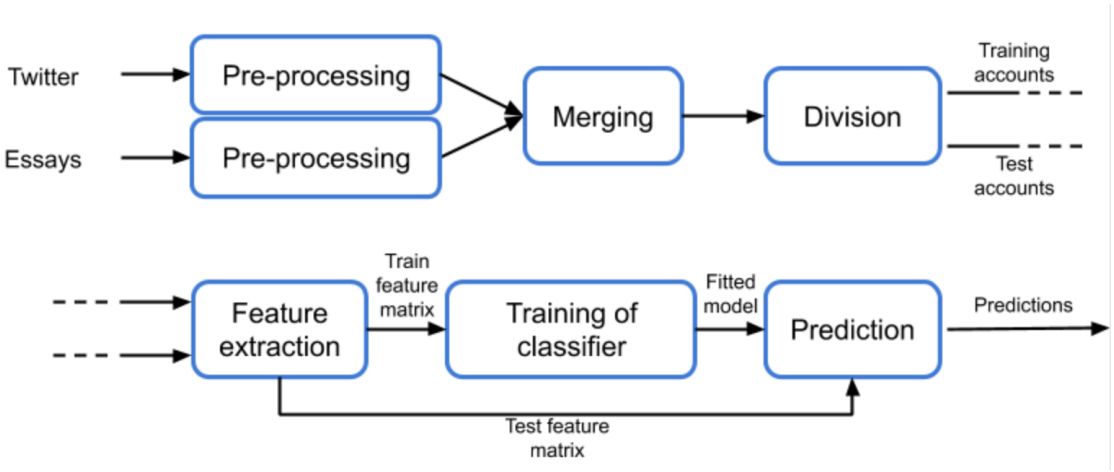

Figure 7.1.: Personality Model

default parameters but testing with different values for the alpha parameter. For the pro-ED experiments, the RidgeClassifier from the same package was used, also testing the alpha parameter and leaving the other parameters at default.

7.2.5. Multi-Layer Perceptron

Multilayer Perceptron (MLP) was chosen as one of the algorithms to test because it is very good at handling high dimensional input data since it can be used for both regression and classification. The algorithm was implemented using sklearn.neural_network.MLPRegressor for the personality detection model, and sklearn.neural_network.MLPClassifier for the pro-ED classifier. As with the other algorithms, all parameters were left at default except for the number of layers which was changed for the experiments.

7.3. Building the Classifiers

As mentioned above, five machine learning algorithms were used in this thesis. All the algorithms were tested in experiments for both the personality detection model and the pro-ED classification model. This section is divided into three parts where the first part covers the personality detection model, the second covers the pro-ED account classifier and the third covers the combination of the two classifiers.

7.3.1. Personality Detection Model

For all the models tested for the final personality detection model, a separate regression model was built for each of the five personality traits, which was also done by Arnoux et al. (2017) with good results. Figure 7.1 displays the layout of the model. The YouTube transcriptions dataset was removed due to it having a detrimental effect on the personality detection. The reasoning behind the removal of the dataset is described in chapter 8.1.

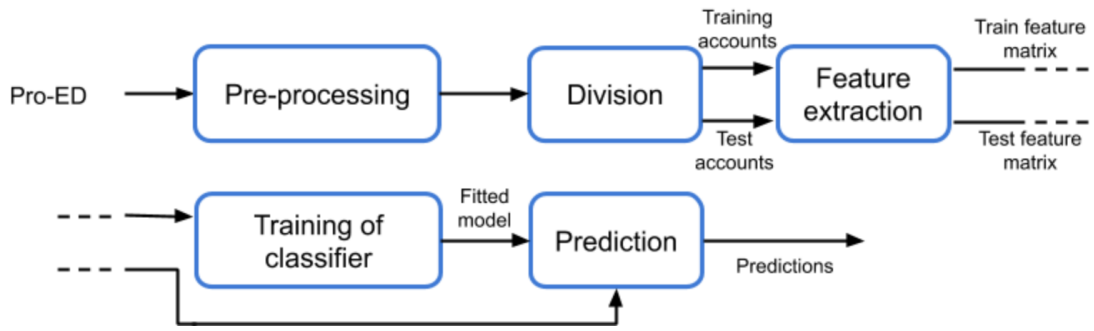

Figure 7.2.: Pro-ED Model

As a result, only the Twitter and Essays datasets were used for the creation of the final personality dataset, which can be seen in the figure. As mentioned in chapter 6.3, the datasets underwent their own pre-processing steps before being merged into one large dataset. This final personality dataset was then divided into two parts: one being a training part, used to train the machine learning algorithm, the second being a testing part used to measure the performance of the model. The test set was left untouched and unseen until the evaluation process. The performance of the model was measured using a Pearson correlation coefficient on the predicted values when applying the models to the test set and the true values of the test set. Part of the reasoning for using this measure was that it was used by Arnoux et al. (2017), who used the same GloVe model. This meant that the results could be compared to see if the models created in this work would produce similar results as the model created by Arnoux et al. (2017).

7.3.2. Pro-ED Classifier

The pro-ED classifier was also built after experimenting with pre-processing steps, feature selection and classification models to create the best possible classifier. This was done by first testing all the chosen features with the SVM classifier, and then testing all the different classifiers using unigrams as the only feature. The classifier with the best results was then applied to the feature set containing the best performing features. Figure 7.2 displays the layout of the pro-ED classification model. As with the personality detection model, the dataset was separated into a training set and a validation set. The various models were compared by how well they performed when being applied to the test set.

7.3.3. Pro-ED Classifier with Personality as a Feature

Since the goal of this thesis was to create a pro-ED classification model with personality as a feature, the personality detection model and the pro-ED classifier described above were combined together into one classification model. Figure 7.3 displays the layout of the final model. As can be seen, the personality model differs from the model described in figure 7.1. The division process is removed in order to be able to use the full personality

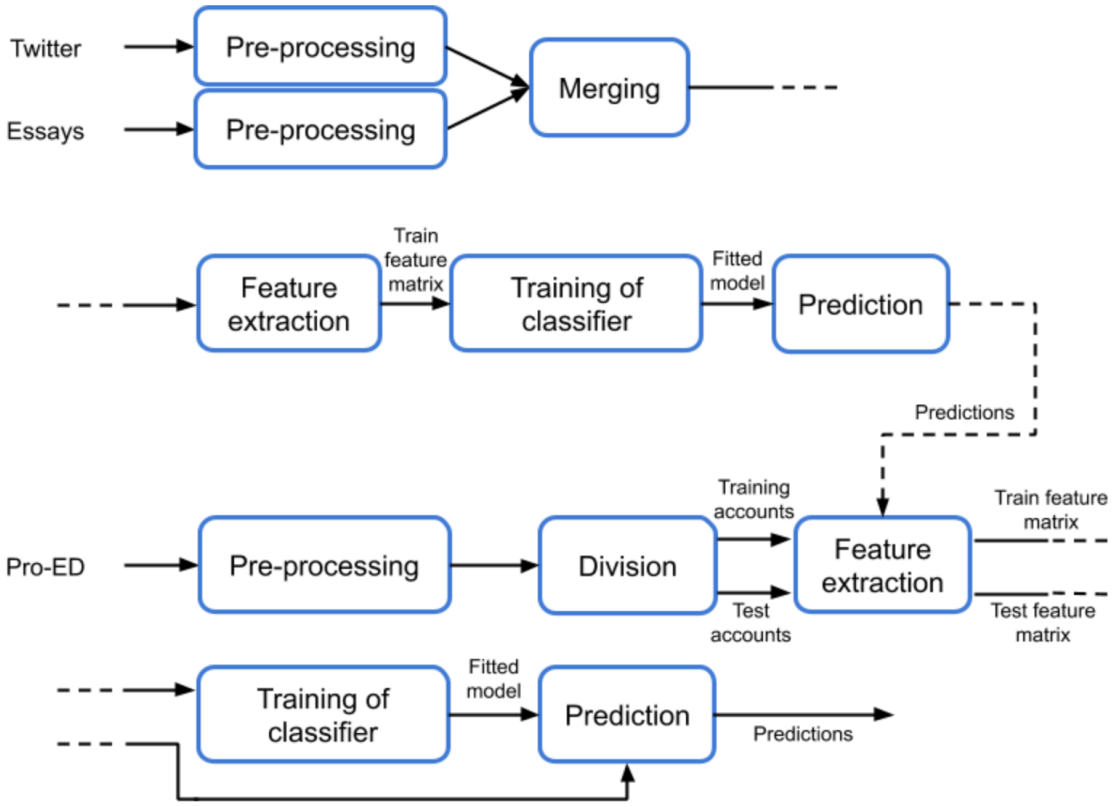

Figure 7.3.: Combined Pro-ED and Personality Model

dataset to train the algorithm. The reason for doing this was that the personality dataset was small to begin with, and by dividing it into a test and training set, it would have been even smaller. Since the performance of the personality model had already been tested, it would have been unnecessary to test it again when being used as a feature, hence the full personality dataset was used in the training of the model. The hypothesis was that this would give better results because of the usage of a larger dataset in the model training. The personality model and the pro-ED model were merged by using the predictions from the personality model as a feature in the pro-ED model, as the figure depicts. The algorithm providing the best personality results and the algorithm providing the best pro-ED results were used to create this final model.

8. Experiments and Results

This chapter covers the experimental plan, experiment setup, and experiment results. All the experiments related to the personality datasets, and the selection and creation of the personality detection model, are presented in the first part of the chapter. The second part contains all the experiments related to the pro-ED dataset and to the selection of the pro-ED classification model. The third and last part of the chapter covers the construction of the final pro-ED classification model.

8.1. Experiments for Personality Detection

The goal of the experiments presented below was to answer research question 1, which focuses on creating the most accurate Big 5 personality detection model as possible. The experiments were divided into three parts, where each part covers an important step in the process of creating the final personality detection model. As mentioned in chapter 6.1.1, the personality datasets all had different structures and composition, but one of them also had Big 5 personality scores that differed from the other two datasets. As a result, one experiment focusing on extra pre-processing was performed with this dataset. The details of this experiment are explained in the first part. All the following experiments were then run on a single, large dataset created from the combination of two of the three personality datasets. Feature selection was an important part of creating an accurate regression model and the experiments involving feature selection are described in the second part. Why only two datasets were used will also be explained in this part. The most promising features were used in experiments focusing on building the best regression model for personality detection. These final experiments are described in the last part. The accuracy of the models was measured using the Pearson correlation coefficient, which is described in chapter 3.3.

8.1.1. Pre-processing Experiments for Personality

Before it would be possible to perform experiments on feature and regression model selection, it was important to make sure that all the personality datasets had the same personality score representation. While the YouTube dataset and the MyPersonality Twitter dataset were annotated in the same manner (floating point numbers ranging from 1-5 for each of the Big 5 personality traits), the MyPersonality Essays dataset was annotated with binary values. Since the Essays dataset was the only one with binary values, while the other two datasets were numeric, it was decided to change the Big 5

Table 8.1.: Results for Binary to Numerical Values Conversion

Conversion method	Extro-version	Agreeableness	Conscientiousness	Emotional stability	Openness
Set Values	0.192	0.306	**0.243**	0.131	0.199
Average Values	**0.195**	**0.415**	0.155	**0.213**	**0.252**

personality trait values for this dataset to numeric values as well. The binary values had the form of *y*, for yes, and *n*, for no.

Two approaches were considered for converting the binary values to numerical. In the first approach, each trait value was assigned a set value for *n* and *y*. The values were set to 2 for *n* and 4 for *y*. These values were chosen based on the data in the Twitter dataset, which contained values from 1-5, and where few people were labeled at the far ends of the spectrum. By choosing 2 and 4, the labels would still belong to the class the person had been assigned, but not be of the unlikely extremes. The second approach was to take advantage of the Twitter dataset having both binary and numerical values, and use the average of the numerical value for each binary value. This meant that for each personality trait, an average numerical value was found for both *y* and *n*. These average values then replaced the binary values in the Essays dataset.

To test which approach would be best, a personality detection experiment was done using Global Vectors (GloVe) with 200 dimensions and Gaussian Process (GP). GloVe with 200 dimensions and GP was chosen because it was reported by Arnoux et al. (2017) to produce good results. The results of this experiment can be seen in table 8.1. The average values from the second approach scored better on most of the personality traits, except for conscientiousness which got a lot worse result. Extroversion stayed roughly the same in terms of Pearson value. For this reason, it was decided to use the second approach for converting the binary values to numerical values.

8.1.2. Establishing a Personality Result Baseline

A personality result baseline was established in order to be able to compare the results of the experiments. To create this baseline, Support Vector Regression (SVR) (regression with Support Vector Machine) was chosen as regression model and Linguistic Inquiry and Word Count (LIWC) was chosen as feature. The reason for using this specific combination was that both had been used a lot for baseline creation in the related research (see chapter 5.2 for more information). All of the 94 available LIWC features were used to create the baseline. A linear SVR model was used as the regression model, with all parameters at their default value. The baseline result can be seen in table 8.2.

Table 8.2.: Baseline for Personality

Extro-version	Agreeableness	Conscientiousness	Emotional stability	Openness
0.153	-0.152	0.024	0.034	0.168

8.1.3. Features for Personality Detection

When running feature experiments for the personality detection model, all the features were tested using both SVR and GP machine learning algorithms. The features tested were word embeddings through GloVe, n-grams, topic models and LIWC. GloVe had shown very good results in combination with GP in related research (Arnoux et al., 2017), and the first experiment focused therefore on testing GloVe with various parameters. The second experiment tested n-grams both on a word level, as unigrams and bigrams, and on character level, with a range of 3-15. The third experiment looked at topic models with a varying number of topics, while the final experiment looked at LIWC with different number of features.

Global Vectors
The first feature experiment focused on testing a pre-trained GloVe model with the four dimension options that were available. This was done in order to see which number of dimensions would produce the best results. The dimension options were: 25, 50, 100 and 200. Naturally, models with higher dimensions become slower, but considering the small amount of data in this dataset, that was not considered a problem. In this experiment, it was predicted that the higher dimensional models would perform better, but it was interesting to see exactly how much would be gained by choosing a larger and slower model.

The results of this experiment can be seen in table 8.3. The best model overall was SVR with 200 dimensions. From the table, it becomes clear that 25 dimensions were too little for some of the traits, both for the GP and the SVR algorithm. This, however, does not seem to be the case for the emotional stability trait. When looking at the higher dimensions, the overall trend is that the model gets slightly better with higher dimensions. In fact, the only case when 200 dimensions were not optimal was with conscientiousness, which had slightly better results with 50 dimensions for GP. In the case of SVR, however, conscientiousness followed the trend of the other traits with gradually increasing scores. It is also possible to see that the GP model performed better than SVR on the experiments with small dimensions, while SVR was best when the dimensions were high. This makes sense knowing that Support Vector Machine (SVM) models often perform well on high dimensional data.

The dataset used for the above results is the dataset created by combining the MyPersonality Twitter dataset and the MyPersonality Essays dataset. The YouTube transcriptions

Table 8.3.: GloVe Results for Different Dimensions

Glove dimension	Algorithm	Extroversion	Agreeableness	Conscientiousness	Emotional stability	Openness
25	GP	0.159	0.406	0.025	0.175	0.168
	SVR	0.120	0.416	0.268	0.158	0.111
50	GP	0.188	0.431	**0.394**	0.185	0.210
	SVR	0.164	0.476	0.300	0.140	0.213
100	GP	0.248	0.452	0.377	0.175	0.230
	SVR	0.265	0.504	0.303	0.136	0.267
200	GP	0.206	0.469	0.390	**0.200**	0.212
	SVR	**0.288**	**0.510**	0.340	0.166	**0.284**

dataset was not included. During the experiments described above, it was discovered that the results of the experiments greatly improved when performed with just two of the three personality datasets. While the results improved slightly on two traits, extroversion and openness, the YouTube transcriptions dataset lowered the results a great deal on the remaining three. The results are presented in table A.1 in appendix A, and show significantly worse scores on agreeableness, conscientiousness and emotional stability. Because of this, it was decided to remove the YouTube dataset completely from further experiments.

N-grams

The second feature to be tested was n-grams. Both unigrams and bigrams were tested on word level, as well as n-grams on a character level with a range of 3-15 as one feature. One regression model was built for each of the five personality traits and n-grams were used as the only feature. All experiments were conducted with both GP and SVR to see how the performance of these two algorithms compared to each other.

Looking at the results in table A.2 in appendix A, it can be seen that the Big 5 personality trait scores are not particularly good for any algorithm in combination with n. With this said, the GP algorithm with n-grams did, in fact, do well on emotional stability when used on a character level. On this particular trait, the GP algorithm actually gave better scores than the best score on the same trait from the GloVe experiments. Compared to the baseline, the scores are also better for agreeableness, conscientiousness, emotional stability and openness. In total, however, none of the values for n outperformed GloVe as a feature. N-grams were therefore not included in the final personality detection model.

Topic Models

The next feature experiment focused on testing topic models as a feature. An LDA Multicore model from Gensim was used with the input being either a Bag of Words (BoW) model or a Term Frequency-Inverse Document Frequency (TF-IDF) model. The

Table 8.4.: Topic Model Personality Results Using SVR

Method	Num topics	Extro-version	Agreeableness	Conscientiousness	Emotional stability	Openness
BoW	5	0.027	**0.066**	**0.126**	-0.019	0.095
BoW	10	0.075	0.064	-0.010	-0.025	**0.117**
BoW	15	**0.131**	-0.042	0.117	-0.033	0.095
BoW	20	0.001	0.060	0.065	-0.009	0.110
TF-IDF	5	0.044	-0.067	-0.027	0.002	-0.027
TF-IDF	10	0.019	-0.007	-0.081	0.086	0.051
TF-IDF	15	0.075	-0.086	0.013	-0.032	0.031
TF-IDF	20	0.075	-0.117	0.010	**0.165**	-0.004

Table 8.5.: Topic Model Personality Results Using GP

Method	Num topics	Extro-version	Agreeableness	Conscientiousness	Emotional stability	Openness
BoW	5	0.044	**0.188**	0.120	0.074	0.128
BoW	10	0.036	0.131	**0.153**	**0.163**	0.120
BoW	15	0.115	0.106	0.040	0.013	0.096
BoW	20	0.111	0.069	0.140	0.112	**0.214**
TF-IDF	5	-0.063	-0.028	-0.026	-0.042	0.096
TF-IDF	10	**0.128**	0.074	-0.019	0.125	0.042
TF-IDF	15	0.082	0.062	0.003	-0.146	0.041
TF-IDF	20	0.074	-0.116	-0.000	-0.065	0.019

experiments looked at different values for the number of topics for each input model. The results were compared using the Pearson value and the experiments were performed using both SVR and GP.

The results of the SVR experiments can be seen in table 8.4 while the results of the GP experiments can be seen in table 8.5. As with the n-grams, the topic models did not produce very promising results with any of the parameters tested. The best topic model results with the use of SVR algorithm were found with BoW and *number of topics=5*. The best overall values can be seen in table 8.5 where the GP algorithm was used. In this table, the results using the BoW input was, with the right number of topics, better than the baseline on all personality traits except for extroversion. For both SVR and GP, the use of BoW produced overall better results than TF-IDF. Still, when comparing these results with the results achieved by the GloVe experiments (table 8.3), they were not very promising. For this reason, Topic Models were also discarded as a feature for the final personality detection model.

Table 8.6.: Personality LIWC Results with GP

Kernel	Extro-version	Agreeableness	Conscientiousness	Emotional stability	Openness
DotProduct + WhiteKernel	-0.073	**0.449**	0.036	**0.511**	-0.109
RBF	**-0.037**	-0.054	**0.039**	-0.012	**-0.048**

Table 8.7.: Personality LIWC Results Using SVR

N	Extro-version	Agreeableness	Conscientiousness	Emotional stability	Openness
35	**0.189**	0.093	-0.007	0.198	0.037
50	0.032	-0.203	**0.078**	-0.287	0.087
60	0.097	**0.255**	0.075	**0.332**	0.134
94	0.153	-0.152	0.024	0.034	**0.168**

LIWC

Linguistic Inquiry and Word Count (LIWC) was tested as the final possible feature for the personality detection model. This experiment was carried out by finding the LIWC scores for each of the documents in the dataset, and then divide this into a training and test set. The feature was tested using the GP algorithm with the Radial Basis Function (RBF) kernel and with a combination of the dot-product kernel and a white-kernel.

Table 8.6 contains the GP experiment results. The results of this experiment are interesting. Whilst the results are poor for extroversion, conscientiousness, and openness, they are far better than any of the previous experiments when it comes to emotional stability. For agreeableness, they are also very good, although the GloVe features did better at this trait. This is interesting because it shows that information about a person's agreeableness and emotional stability can be found from LIWC features, while other personality traits are harder to detect.

The feature was also tested using SVR. This made it possible to find the n most important features. By choosing only the most important features noise can be reduced, which leads to higher accuracy. Because of this, several values for the number of features were tested to find the optimal number for this specific task. The best experiment results can be seen in table 8.7 (the complete experiment results can be seen in table A.3 in appendix A). These results are, in total, better than those achieved by GP. Still, for agreeableness and emotional stability, the two traits where GP did very well, the SVR model did not produce quite as high scores. Compared to the other traits, the scores for these two traits are still the highest out of all the traits with the SVR algorithm. This gives further support to the theory that these two traits are easier to estimate with the LIWC features. The results all over are still better than the baseline results for all the traits. They did

Table 8.8.: Personality Results for Different K-Values Using K-NN

k	Extro-version	Agreeableness	Conscientiousness	Emotional stability	Openness
2	0.001	0.012	0.056	**0.142**	0.122
5	0.036	0.050	0.018	0.098	0.180
8	0.050	0.069	0.015	0.121	0.195
10	0.060	0.052	0.007	0.130	0.193
15	**0.064**	**0.110**	**0.069**	0.112	**0.218**

not, however, compare to the GloVe feature results and were therefore not included in the final personality detection model.

8.1.4. Regression Models for Personality Detection

All the machine learning algorithms described in chapter 7.2 were tested in order to build the personality detection model. Two of these algorithms stood out in the related research as the best when it came to personality detection, namely SVM and GP (see chapter 5.2). Because of this, these two algorithms were both tested more extensively than the others. While the remaining algorithms were tested using Global Vectors (GloVe) as the only feature, both SVR and GP were tested on all of the features with different configurations. The SVR and the GP algorithms were tested first, followed by the K-Nearest Neighbors (k-NN) algorithm, the Ridge Regression (RR) algorithm and finally the Multilayer Perceptron (MLP) algorithm.

Support Vector Regression and Gaussian Process
The SVR and GP models were put up against each other in all the experiments that tested features for the personality detection model. The results from these experiments can be seen in the tables presented in chapter 8.1.3. The best overall performance of all the experiments was found using SVR with Global Vectors (GloVe) at 200 dimensions as the feature. Compared to the baseline, this experiment got much better scores on all of the personality traits. As a result, GloVe with 200 dimensions was used for the remaining regression model experiments. This feature was chosen as it was the one that had produced the best performance overall with both SVR and GP.

K-Nearest Neighbors
K-Nearest Neighbors (k-NN) was used with default values and experiments focused on using different values for k. The results can be seen in table 8.8. While the model produced somewhat acceptable results on openness, the overall results were not very good, especially when compared to the results achieved by SVR and GP. It should be noted that the results got better as k increased, and it is possible that even better scores could have been produced with higher levels for k than what was tested in these experiments.

Table 8.9.: Personality Results with Different Alphas for Ridge Regression

Alpha	Extro-version	Agreea-bleness	Conscien-tiousness	Emotional stability	Openness
0.001	0.162	0.299	0.119	0.120	0.186
0.01	0.163	0.317	0.129	0.127	0.193
0.1	**0.172**	0.375	0.148	0.171	0.225
1	**0.172**	**0.430**	**0.154**	**0.257**	**0.227**

Table 8.10.: Personality Results with Different Hidden Layer Sizes Using MLP

Hidden layer size	Extro-version	Agreea-bleness	Conscien-tiousness	Emotional stability	Openness
50	0.198	0.090	0.022	0.371	0.155
100	**0.207**	0.122	0.086	**0.379**	0.157
200	0.198	0.249	0.109	0.197	0.184
500	0.184	**0.256**	**0.111**	0.190	**0.235**

Ridge Regression
Ridge Regression (RR) was tested with default values except for the alpha parameter, which was tested for different values. The results of this experiment can be seen in table 8.9 and are promising compared to the k-NN model. Compared to the SVR and GP algorithms, the results are noticeably better on emotional stability, but on the other Big 5 personality traits, they are still not as good, making both SVR and GP better overall choices.

Multilayer Perceptron
The experiments with Multilayer Perceptron (MLP) all used default values and the algorithm was tested with different sizes for the hidden layer. The results from this experiment can be seen in table 8.10. The results were promising, but not as good as either GP or SVR. However, it should be noted that the results on openness were far better than both GP and SVR for layer sizes of 50 and 100.

8.2. Experiments for Pro-ED Classification

The experiments related to the creation of the pro-ED classification model are divided into four focus areas, all aiming to answer the second research question. The first focus area contains the pro-ED dataset pre-processing experiments. These experiments were done to make sure that the pre-processing steps would be beneficial and not harmful to the final results of the pro-ED classifier. Focus area two contains experiments related to the establishment of a pro-ED result baseline. Establishing a result baseline was important as the two following focus areas would be measured up against this baseline. The experiments in the third focus area focus on feature selection and tuning, while the

Table 8.11.: Pre-Processing Experiment Results of Pro-ED Dataset

Pre-processing	Precision	Recall	F1
No pre-processing	0.997	0.978	0.988
Decode	0.997	0.979	0.988
Decode + remove remaining hex	0.997	0.979	0.988
Decode + remove remaining hex + s,m,t	0.997	0.979	0.988
Decode + remove remaining hex + s,m,t + convert *amp* to *and*	0.997	0.979	0.988
Decode + remove remaining hex + s,m,t + convert *amp* to *and* + remove numbers	0.997	0.980	0.988

fourth and final focus area contains classifier selection experiments.

Precision, recall and F_1 scores are calculated for all the following experiments. When only two labels were used, as was the case for most of the experiments, the binary calculation was used, regarding pro-ED as the positive label. In the multi-class experiments, the chosen approach was to use the weighted option of a macro average calculation as this accounts for imbalances in the classes.

8.2.1. Pre-processing Experiments for Pro-ED

Five out of eight pre-processing steps, described in chapter 6.4, were tested in order to decide whether they would be included in the pre-processing of the pro-ED dataset. The pre-processing steps were all intended to deal with inconsistencies in the pro-ED dataset. Experiments were done to see how each of the pre-processing steps would affect the performance of the classifier. The remaining three pre-processing steps not experimented on were pre-processing steps that, in the authors' opinion, did not threaten the quality of the dataset. Experiments on these steps were, therefore, not necessary. The experiments were executed by applying the pre-processing step on the tweets in the dataset. This means that bio and account display name were kept unprocessed for the tests. This was done in order to save time. The pre-processed dataset was then used as input for an Support Vector Machine (SVM) algorithm with a feature set consisting of unigrams, bigrams, username, bio, and emojis. A combination of all five features was also tested. The features were used as input for a 10 fold cross-validated SVM that would classify the accounts to decide whether they were pro-ED or not. The reason for using SVM was that it would be used for baseline creation in the next experiment phase. The dataset potential was measured through precision, recall and F_1 scores. These measures were calculated for all experiments and the results can be found in table 8.11. The pre-processing steps and how they were run is described below.

Decode non-ASCII characters
As described in chapter 6.4, a lot of tweets in the dataset were represented in an encoded

non-ASCII format. That is, they were represented as hex encoded byte strings instead of the original symbols. All tweets in the pro-ED dataset were run through a Latin1 encoder and a UTF-8 decoder in order to decode the non-ASCII tweets. The Latin1 encoder was used with the only purpose of making sure that the partly processed byte string had the right format. An experiment was done to see how this affected the dataset and the results can be seen in table 8.11.

Delete Remaining Encoded Strings
Because some accounts had numbers removed as a result of the modifications done by Giæver (2018), some special characters could not be decoded in the previous pre-processing step. This led the dataset to contain a large number of words like *xb*, *x* and *xe* which introduced a lot of noise. Since these characters could not be decoded, an experiment was done where a Regular Expression (RegEx) was used to remove all instances of *x* standing alone or *x* followed by a letter in the range of a-f with spaces before and after. This would also remove some words that were meant to be there, such as the much used emoji *xD*, which is often used as an emoji in informal written text, or *x*, sometimes meaning *kiss*. The experiment results can be seen at the end of this section in table 8.11.

The effect of this pre-processing step became very clear when creating a topic model visualization from the dataset. As seen in figure 8.1 the topics, before the decoding and deletion of the byte strings, included words like *xd*, *xb* and so on. These took up quite a large part of the features without bringing much useful information, and actually held the spots for the three most important terms in the topic model. The relation between the topics and the distribution can also be seen to overlap quite a lot, making it difficult to distinguish between the topics. Figure 8.2, on the other hand, shows the most salient topics after the decoding and removal of the byte strings. The changes are quite noticeable with the most significant words changing from *xd*, *xb*, and *xe*, to *thinspo*, *disord* (from disordered), and *skinni*. There is also less overlap between the topics and the distribution of the topics is wider. This means that the topics have become easier to distinguish from each other.

Merge Single Letters with Associated Words
Due to the dataset modifications three characters, *s*, *m* and *t*, were in many cases separated from the word they belonged to, probably as a result of replacing symbols with space characters. In order to fix this, the characters were run through a RegEx. The RegEx would detect all occurrences of any of these three letters standing preceded by a space and followed by any whitespace character. When a match was found, the whitespace character before this letter would then be removed, leading to merging it with the previous word. An experiment was performed to see how this letter merge affected the dataset and the results can be seen in table 8.11.

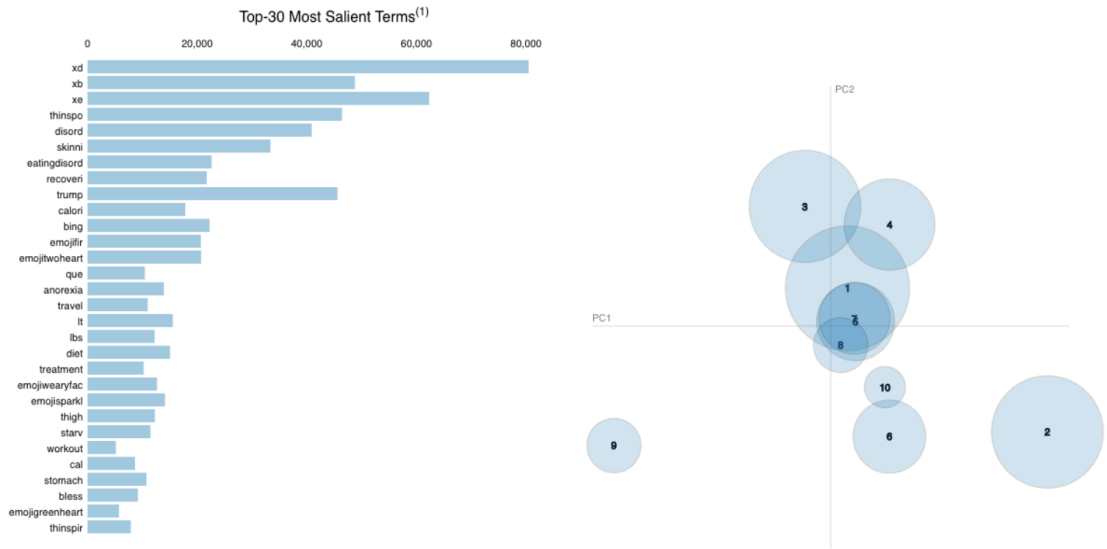

Figure 8.1.: Most Relevant Topics Before Removing Byte Strings

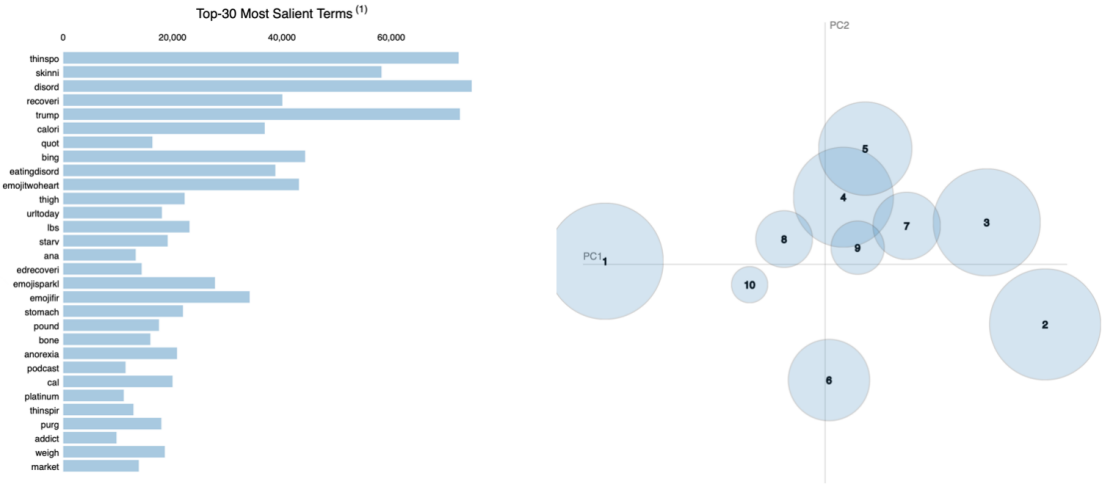

Figure 8.2.: Most Relevant Topic After Removing Byte Strings

Replace *amp* with *and*
As described in chapter 6.4, the character & in tweets was represented as *amp*. An experiment was carried out by replacing *amp* with *and* to see if that would make any difference. The hypothesis was that this would have an impact on the dataset potential because the word *amp* has no meaning to any of the pre-trained Natural Language Processing models. Models like Part of Speech taggers, LIWC and all models making use of stop words would not recognize this word, while the word *and* is likely to be present in all of these. To replace the word, a RegEx was used that would detect the word *amp* surrounded by whitespace characters before and after. The word, when detected, was then replaced by the word *and*. The results of the replacement can be seen in table 8.11.

Delete Numbers
Another problem with the dataset was that some Twitter accounts had had numbers removed from their tweets while others had not. This introduced an inconsistency in the dataset, and an experiment was done to see if removing the numbers from all accounts would improve the classification. The removal was done by running the tweets from each account through a RegEx that filtered out all numbers. The results from the experiment can be seen in table 8.11.

Pre-processing Experiment Results
Table 8.11 shows the results of running each of the pre-processing steps through an SVM algorithm with the combined features mentioned above. These results are quite similar for all tests, which were to be expected seeing as the classifier was already very accurate. However, the results for recall did get slightly better with more pre-processing steps added to the algorithm. Considering how the pre-processing gave slightly better and better results on this metric, whilst keeping precision and F_1 the same, it was decided to keep all the steps. This was because the pre-processing was considered to be correcting some faulty modifications that were done to the original tweets, and bringing them closer to their original form. By adding in these steps, it is natural to believe that new data will be more correctly classified as they will have a form more similar to the pre-processed data, than to the modified tweets in the dataset that was obtained from Giæver.

8.2.2. Establishing a Pro-ED Result Baseline

In the research done by Giæver (2018) the Support Vector Machine (SVM) algorithm provided the most accurate results. This finding was also supported by a number of other research articles (see chapter 5). The result produced by the state-of-the-art pro-ED classification model proposed by Giæver is the focus for improvement in this thesis, and because of this, SVM was chosen as the machine learning algorithm that would be used to create a baseline for result comparison. In order to create a trustworthy baseline, it was important to make sure that the SVM experiment proposed by Giæver was possible to recreate and that the results proved to be similar. This would strengthen the credibility of Giæver's results as well as provide a valid starting point for the baseline creation in this thesis. As mentioned in chapter 6.2.1, the dataset used by Giæver was re-annotated

Table 8.12.: Baseline Results with Giæver's Recreated Experiment

Features	Authors' Pro-ED			Giæver's Pro-ED		
	Precision	Recall	F_1	Precision	Recall	F_1
Unigrams	0.995	0.975	0.985	0.982	0.974	0.978
Bigrams	0.993	0.960	0.977	0.985	0.963	0.974
Username	0.883	0.782	0.829	0.910	0.741	0.817
Bio	0.903	0.759	0.824	0.880	0.745	0.807
Emojis	0.860	0.779	0.817	0.871	0.806	0.837
Combined	0.997	0.976	0.987	0.982	0.978	0.980

to make sure that the data was as correctly labeled as possible, but in order to recreate her results, it was decided that the original dataset (prior to re-annotation, and the one Giæver used herself) would be used.

Thus the first baseline experiment consisted of running the pro-ED dataset, prior to re-annotation, through an SVM. The five feature groups Giæver used in her experiment (unigrams, bigrams, username, bio, and emoji) were tested individually with the SVM model as well as combined. Five fold cross-validation was used to prevent overfitting. One test was performed for each feature and one was performed for the combined features, resulting in six results. The results are presented in table 8.12 together with the results Giæver got from her experiment. Compared to each other, the results prove to be similar, meaning that it was indeed possible to recreate the experiment and get similar results. These results show, in line with the observations made by Giæver, that out of the five feature groups unigrams gave the best results while the combined model was the best overall. Because the combined model produced the best overall result, it was decided to use the combined model as the baseline starting point.

As mentioned above, the dataset used to create the baseline starting point was the dataset Giæver used in her experiments. However, this was not the dataset that would be used in future experiments in this thesis. The dataset that would be used had both been re-annotated and pre-processed (as described in chapter 6.2.2 and 6.4) and was therefore different in structure and composition to the one used to create the baseline starting point. Since the results produced by experiments ran on the re-annotated and pre-processed dataset would be compared to the result baseline, it was decided that the result baseline should come from the re-annotated and pre-processed dataset. Two more experiments were therefore conducted: one for the re-annotated dataset and one for the re-annotated and pre-processed dataset. The goal of these two experiments was first and foremost to create a baseline that could be used for result comparison, but also to see how the changes done to the dataset would affect the results when used with the features and classifier that created the baseline starting point.

In order to test the re-annotated pro-ED dataset, it was run through the same experiment

Table 8.13.: Baseline Results with Re-Annotated Dataset

Features	Precision	Recall	F_1
Unigrams	0.994	0.978	0.986
Bigrams	0.994	0.965	0.979
Username	0.882	0.783	0.83
Bio	0.904	0.759	0.825
Emojis	0.861	0.782	0.819
Combined	0.997	0.978	0.988

Table 8.14.: Baseline Results with Pre-Processed and Re-Annotated Dataset

Features	Precision	Recall	F_1
Unigrams	0.995	0.984	0.990
Bigrams	0.996	0.968	0.981
Username	0.884	0.784	0.831
Bio	0.910	0.757	0.826
Emojis	0.866	0.799	0.831
Combined	0.997	0.982	0.989

that created the baseline starting point. The same feature groups were used to make sure that the results could be compared. Table 8.13 contains the results of the experiment. Unigrams, bigrams, and emoji features show better results on both precision, recall and F_1 scores compared to before the re-annotation, while username and bio show slightly poorer results. The combined results, on the other hand, show better results in both precision and F_1, but slightly poorer recall.

The final baseline experiment focused on running the re-annotated and pre-processed pro-ED dataset through the same experiment, with the same features, as before. This was the most important baseline experiment as this was carried out with the final version of the pro-ED dataset, which would be used for all following experiments. Table 8.14 contains the results of the final baseline experiment. All the features showed improvement in both precision, recall, and F_1 score, except for unigrams which had a slight decrease in precision. The combined features showed improvement in both recall and F_1 score, while precision stayed the same as the results from the re-annotation. When compared to the baseline starting point, there is an improvement in precision, recall and F_1. The baseline for this thesis will be represented through these final results and all future experiments will be compared to this.

Even though the baseline was established, another experiment was conducted in order to be able to see how the baseline would look if applied to a multi-class dataset. Hence, the baseline experiment described above was also conducted using all the labels in the dataset: pro-ED, pro-recovery and unrelated. While the other experiments only used a

Table 8.15.: Baseline Results with Pre-Processed and Re-Annotated Multiclass Dataset

Features	Precision	Recall	F_1
Unigrams	0.971	0.971	0.970
Bigrams	0.967	0.966	0.965
Username	0.816	0.821	0.810
Bio	0.864	0.860	0.857
Emojis	0.815	0.825	0.813
Combined	0.976	0.976	0.975

binary classification of pro-ED vs non-pro-ED (pro-recovery and unrelated labels were combined into non-pro-ED), it was decided that it would be interesting to also test how the algorithm performed with all three classes.

The results of this experiment can be seen in table 8.15. Both precision, recall and F_1 score (calculated as a weighted average of the macro scores for each label) showed poorer results than the binary baseline results. The reason for this is discussed in chapter 9.1.9. The final pro-ED classification model will be run as a multi-class as well as a binary class in order to see how the coming experiments have affected the results.

8.2.3. Features for Pro-ED Detection

All features mentioned in chapter 7.1 were tested to see if they could provide useful information for the final pro-ED classifier. SVM was used for all the feature experiments in order to keep the features the only thing changing between the experiments. SVM was chosen because it was the best performing algorithm in Giæver (2018), as well as being the algorithm used for the baseline in this thesis. The features tested for the pro-ED classifier were n-grams, Global Vectors, topic models, Part of Speech and Linguistic Inquiry and Word Count.

N-grams
Three feature experiments were conducted with n-grams. The first two experiments tested unigrams and bigrams with varying limits for the number of features. The last experiment tested the performance of trigrams with the lowest limit. For all experiments, all other parameters were left at default. The best results of the experiments can be seen in table 8.16. The results of each individual experiments can be found in appendix A.

As seen in table 8.16 the results from the unigrams experiment show that the best performance was achieved by using 1200 features. Unigrams also produced the best overall results compared to bigrams and trigrams. In table A.4 in appendix A, the complete results for the unigram experiments testing different values for the maximal number of features are presented. Looking at the results from the complete experiment, it can be seen that the F_1 score and precision increased from 300 to 900, and then stayed

Table 8.16.: Pro-ED N-Grams Feature Results

Type	Features	Precision	Recall	F_1
Unigrams	1200	0.995	**0.986**	**0.990**
Bigrams	5000	**0.996**	0.971	0.984
Trigrams	300	0.966	0.895	0.929

Table 8.17.: Pro-ED Glove Feature Results

GloVe dimension	precision	recall	F_1
25	0.969	0.959	0.964
50	0.977	0.966	0.972
100	0.986	0.968	0.977
200	**0.987**	**0.970**	**0.978**

the same for all the experiments. With recall, however, the results increased until 1200, and then decreased slightly as the maximal number of features increased. This can be because 300 and 900 features are too little and do not give enough information, while 2000 or more features might start to give too much information. This can then make the model overfit to the training data and therefore not perform as well on test data.

With the bigrams, it can be seen from table 8.16 that the optimal value for the number of features tested was 5000. Looking at the complete experiment results (table A.5 in appendix A) it can be seen similar trends as with the unigrams. The measurement scores for all three metrics gradually improved up to the use of 5000 features, after this the scores worsened.

For trigrams, only one value for the maximum number of features was tested, namely 300. This was due to the experiments taking a very long time to run. The experiment results in table 8.16 showed that trigrams performed worse than the two other values for n. In the full experiment result tables for unigrams and bigrams (table A.4 and table A.5 in appendix A) it can be observed that the results with a lower limit for the number of features also got worse from unigrams to bigrams. It could be the case that trigrams would perform better with a larger number of features. However, because of the time it took to run this experiment, no further values were tested. Looking at the results from this experiment it was decided that pursuing this feature would most likely not lead to any improvements in the results beyond what could be found from unigrams and bigrams. Trigrams were therefore not included as a feature for the final classifier.

Global Vectors

As with the personality dataset Global Vectors (GloVe) was tested, using all the different dimensions, on the pro-ED dataset as well. The results from this experiment can be seen in table 8.17. Improvement was made on all metrics up until 100 dimensions, while

Table 8.18.: Pro-ED Topic Model Feature Results with Different Number of Topics

Method	Num topics	Precision	Recall	F_1
BoW	10	**0.989**	**0.990**	**0.989**
TF-IDF	10	0.972	0.989	0.980

only marginally improving from 100 to 200 dimensions. The best results were achieved by using 200 dimensions. Overall, GloVe showed good results as a standalone feature, outperforming username, bio and emoji features from the baseline. Still, it did not perform as good as the unigrams and bigrams and was therefore not included in the final feature set.

Topic models
The feature experiment with topic models looked at changing the number of topics generated while keeping the other parameters at default. The dictionary was set to contain at most 10 000 words and was set to not keep words that were present in less than 20 accounts or in more than 50% of the accounts. Both a BoW model and a TF-IDF model were tested as input to the topic models. In this experiment, the BoW corpus with 10 topics turned out to be best for all three metrics, as seen in table 8.18. 10 topics also produced the best result for TF-IDF, but as can be seen, the results are not nearly as good as for BoW. The complete experiment results can be seen in table A.6 in appendix A. These results show that the BoW corpus performed better than the TF-IDF corpus with the same number of topics on all tests.

Looking at these results, it can be concluded that topic models is a very promising feature for pro-ED detection. With an F_1 score of 0.989 it is in fact as good as the baseline with all features combined. For this reason, topic models were included in the final classifier, with the BoW corpus as the input and the same settings for the dictionary.

Part of Speech tagging
The Part of Speech (POS) feature experiments were carried out by parsing all the combined tweets for each Twitter account through the POS tagger mentioned in chapter 4.1. In order to use the POS tags as a feature, the tags were considered as an n-gram model on character level. The experiment conducted on this feature looked at using different values for n in the n-grams, meaning it would look at different length sequences of tags. The results of this experiment can be seen in table A.7 in appendix A, and show that the results improve for n values up to five, before worsening with larger n values. The optimal n value was five. The results were, however, not good enough to be added to the final classifier.

LIWC
Linguistic Inquiry and Word Count (LIWC) analysis as a feature was tested using the LIWC 2015 version (Pennebaker et al., 2015). The experiment was carried out using a

Table 8.19.: Pro-ED LIWC Feature Results

n	Precision	Recall	F_1
5	0.951	0.964	0.931
10	0.967	0.955	0.844
15	0.982	0.946	0.965
20	0.967	0.968	**0.971**
25	0.936	0.964	0.951
30	0.971	0.983	0.970
35	0.972	0.964	0.957
40	0.976	0.968	0.967
50	0.918	0.946	0.954
60	0.962	0.973	0.935
70	**0.984**	0.978	0.933
80	0.970	**0.985**	0.968
90	0.932	0.922	0.819
all	0.878	0.962	0.847

different number of features from the LIWC feature set. The features chosen were always the n most important according to the Support Vector Machine (SVM) coefficients. The results from this experiment can be seen in table 8.19. The best results were found when 20 of the most important features were used, but since the results were not as good as the baseline F_1 score LIWC as a feature was not included in the final classifier.

8.2.4. Classifiers for Pro-ED Detection

For the machine learning algorithm selection experiments, the Twitter accounts in the training set were further split into a training and a validation group. The validation group consisted of 20% of the accounts from the original training set. The accounts that were in the training group were used to fit a TfidfVectorizer, which was then used to transform the tweets from both groups into unigram features. The unigram was chosen as the feature for these experiments because it had shown good results in the feature experiments presented above. Unigrams was also inexpensive to compute, making the experiments faster. Each machine learning algorithm was then trained on the unigrams from the training group, and then evaluated using the unigrams from the validation group. The metrics used to evaluate the accuracy of the algorithms were precision, recall and F_1 scores. The first algorithm to be tested was the Support Vector Machine algorithm, which had been used to create the result baseline. The second algorithm was the Gaussian Process algorithm, which had shown promising results in related research on personality (see chapter 5.2). The third algorithm was the K-Nearest Neighbors algorithm, while the fourth and fifth algorithms were Ridge Regression and Multilayer Perceptron.

Table 8.20.: Pro-ED SVM Result with Different Kernels

Kernel	Precision	Recall	F_1
RBF	0.988	0.868	0.924
Linear	**0.990**	**0.995**	**0.992**
Sigmoid	0.983	0.770	0.864

Support Vector Machine
The first experiment on the Support Vector Machine (SVM) classifier tested different values for the C parameter. The C parameter is the error term that decides how much to punish wrong classifications in the training phase. The values tested ranged from 0.01 to 10, and the kernel used was the linear kernel. The results, which are presented in table A.8 in appendix A, show very similar results for all values above 1, but a slight increase in recall all the way. Precision was slightly better at *C=1.5* and *C=2*, but was fairly similar for all values from 0.1 and over. The best score for F_1 was achieved at *C=2* and *C=10*. Out of those two, *C=2* was chosen to be used in the final classifier.

The second experiment with SVM tested the use of different kernels to see which would give the best results. The kernels tested were RBF, the linear kernel, and the sigmoid kernel. For the other parameters, gamma was set to *scale* while all other parameters were left at the default value. The results in table 8.20 show that the linear kernel performed best, with the RBF as the second best and the sigmoid kernel as the worst. Overall, all three kernels performed fairly well. Looking at the results for the precision and recall metrics, it also becomes clear that while all three kernels performed very good at precision, the non-linear kernels did not perform as well as the linear on recall. This means that the non-linear kernels will have marked fewer accounts as pro-ED in total, but that the ones that were marked as pro-ED were almost always correct.

Gaussian Process
With the Gaussian Process (GP) algorithm experiments, two different kernels were tested, both with the default parameters. The kernels tested were dot-product, in combination with the white-kernel, and the RBF kernel. The results in table A.9 in appendix A were all very good, with F_1 scores of 0.967 and 0.971. They were, however, not as good as some of the other algorithms. The model was also very slow to train compared to the rest, which was another drawback. As a result, the GP algorithm was not considered for the final classifier.

K-Nearest Neighbors
For the K-Nearest Neighbors (k-NN) experiments, different values for k were tested. The other parameters were kept at default. The k-values tested were 1, 2, 5, 8, 10 and 15 and the experiment results can be seen in table 8.21. The results show a gradual improvement of the performance metrics as the k-value increases until it reaches 10. With *k=15* the performance scores decrease. Overall, the results are very promising, but not as good

Table 8.21.: Pro-ED K-NN Results with Different K-Values

K	Precision	Recall	F_1
1	0.948	0.982	0.964
2	0.967	0.972	0.969
5	0.962	**0.987**	0.974
8	0.969	0.984	0.977
10	**0.972**	0.984	**0.978**
15	0.967	0.984	0.976

Table 8.22.: Pro-ED Ridge Regression Results with Different Alpha Values

Alpha	Precision	Recall	F_1
0.001	0.982	0.964	0.973
0.01	0.987	0.972	0.979
0.1	0.987	**0.979**	0.983
1	0.992	0.977	**0.984**
2	0.992	0.966	0.979
4	**0.995**	0.951	0.972
6	0.992	0.943	0.967

as with linear SVM or the MLP (described below). It can also be observed that recall stayed fairly similar while precision had larger changes with different values for k.

Ridge Regression

As with the other experiments, the Ridge Regression (RR) experiment was conducted using all default parameters except for the alpha parameter, which was the one being tested. The results from this experiment can be seen in table 8.22. The best F_1 score (0.984) was achieved when *alpha=1*, which is better than both the k-NN and GP classifiers. Compared to the SVM classifier, however, it is still slightly worse.

Multilayer Perceptron

For the Multilayer Perceptron (MLP) experiments all parameters, except for the hidden layer size, were set to default. The experiment tested different hidden layer sizes, using the values 50, 100, 200, 500. The results from this experiment are presented in table 8.23. The results were very good for all layer sizes tested. All the different scores for all the hidden layer sizes were at or above 0.99, and for recall, some even got 100%. This made the MLP algorithm the best performing algorithm out of the five tested, which was why it was chosen as the algorithm to test for the final classifier.

Table 8.23.: Pro-ED MLP Results with Unigrams and Different Layer Sizes

Layer size	Precision	Recall	F_1
50	**0.990**	**1.0**	**0.995**
100	**0.990**	0.997	0.994
200	**0.990**	**1.0**	**0.995**
500	**0.990**	0.995	0.992

8.3. Building the Final Classifier

Through the algorithm experiments described above, it was discovered that, all traits considered, the best performing algorithm and feature for personality detection was the Support Vector Regression (SVR) algorithm with Global Vectors (GloVe) as the feature. The algorithm that produced the best pro-ED classification results was the Multilayer Perceptron (MLP) algorithm. In the following experiments the personality detection model, created with the optimal personality algorithm and feature, will be included as a feature in the pro-ED classifier in order to create the final pro-ED classification model. Support Vector Machine (SVM) was also tested as an algorithm for the final pro-ED classifier because it was used in the baseline and would be able to show how the new feature set performed compared to the baseline. Two different experiments were run with both pro-ED algorithms, one having personality as the only feature for pro-ED detection, the other having personality as one of the features competing for a spot in the pro-ED classification algorithm.

8.3.1. Pro-ED with Personality as Only Feature

The reason for running this experiment was to see exactly how personality as a feature affected the performance of the pro-ED classification algorithm. The algorithm used for the pro-ED classifier was the SVM algorithm. Using this algorithm meant that it would be possible to compare the results directly to the baseline. The personality feature was created with the SVR algorithm. This experiment was performed using only the training dataset and with 5-fold cross-validation, in the same manner as the feature and classifier selection experiments. The results from the experiment produced a precision score of 0.954, a recall of 0.961 and an F_1 score of 0.957. This is very good considering the personality detection model had not been trained on the pro-ED dataset, only tested. They were, however, not as good as the baseline.

8.3.2. Pro-ED with Personality as Part of a Feature Set

While personality in itself is interesting as a feature, it is also interesting to see how well personality affects the classifier when part of a larger feature set. To determine which features would remain in the final classifier, the F_1 scores of each feature was considered. The lower limit for a feature to be included was set to 0.98, as this was the F_1 score of the experiment conducted by Giæver (2018). Any feature with F_1 score below the lower

Table 8.24.: Final Classifier Results

Algorithm	Features	Precision	Recall	F_1
MLP	with personality	0.995	**0.984**	**0.990**
SVM	with personality	**0.998**	0.979	0.988
MLP	without personality	0.993	0.982	0.987
SVM	without personality	0.995	0.982	0.988

Table 8.25.: Final Classifier Multi-Class Results

Algorithm	Feature	Precision	Recall	F_1
MLP	with personality	**0.976**	**0.976**	**0.975**
SVM	with personality	0.972	0.972	0.972
MLP	without personality	0.974	0.974	0.974
SVM	without personality	0.972	0.972	0.972

limit was discarded and therefore not included in the final classifier. Three features had F_1 scores above the limit, namely unigrams, bigrams, and topic models.

It was decided that two algorithms would be tested for the final pro-ED classification model. MLP, the algorithm with the highest F_1 results, was chosen as the first algorithm. SVM was chosen as the second. The reason for choosing to include SVM was that it would be interesting to compare the effect of the new feature set with the baseline (which was created with the SVM algorithm).

This final experiment was set up with the three most important features being used together on the MLP and SVM algorithms. The test data that had been taken out before the feature and algorithm tests were used to test these classifiers. Both algorithms were tested with the feature set, including and excluding the personality feature created by the personality detection model. All tests were performed 5 times to find the average values for the scores. The results from this final experiment can be seen in table 8.24.

As seen in table 8.24, both algorithms performed well on the test data. The MLP algorithm performed slightly better than the SVM on the F_1 score when personality was included as a feature. The opposite was true when personality was excluded. In both cases, SVM performed slightly better on precision, while MLP performed better on recall. MLP was better when personality was included and equal to SVR when it was not. It can also be noted that the precision increased with both algorithms when personality was included, while recall increased for the MLP classifier, but decreased with SVM. In addition to these experiments, tests were also done with this final classifier where each feature was tested alone. The results for these tests (described in table A.10 in appendix A) were all poorer than the combined feature set.

The final pro-ED classifier was also tested with a multi-class output where the labels pro-ED, pro-recovery and unrelated were used. For this experiment, the features were the same as the final classifier and the feature set was used with and without personality. The results can be seen in table 8.25. These results show that the MLP algorithm with a feature set including personality produced results identical to the multi-class baseline results. For all the other combinations of classifier and feature set, the score is slightly lower than the baseline.

9. Evaluation and Discussion

This chapter presents an evaluation of the research done in this thesis. The evaluation is divided into three parts, where the first part contains a discussion concerning the research process and results. The second part covers the ethical aspects of handling sensitive data and doing research on a mental illness. The third and last part covers the limitations that affected the quality of the research.

9.1. Discussion

The research process and the results it yielded are discussed below. The discussion is divided into several discussion areas. Each discussion area covers an experiment presented in chapter 8 and is therefore named the same as the experiment. This is in order to easier understand which experiment the discussion concerns.

9.1.1. Removal of the YouTube Dataset

As a part of running feature experiments for the personality detection model, it was discovered that running the experiments without the YouTube transcriptions dataset yielded better results than what it did with the dataset. It is intuitive to think that having a larger dataset to train the personality detection algorithm would improve the results, but this was not the case for the YouTube transcriptions dataset. One reason for the poorer results might be that the YouTube dataset stemmed from a different source than the other two datasets. The Big 5 personality trait scores might therefore have been calculated based on a different score estimation. It could also be that the linguistic form was different for the YouTube transcriptions dataset, compared to the other two datasets, because the transcription in the YouTube dataset was created from oral speech and not written text. The YouTube dataset contained 400 transcriptions so the removal did not diminish the personality dataset too much. Because of this, the dataset was removed.

9.1.2. Features for Personality Detection

The feature experiments for the personality detection model were carried out using both the Support Vector Regression (SVR) algorithm and the Gaussian Process (GP) algorithm. The reason for using both of these machine learning algorithms was that the experiments usually ran very quickly, due to the small dataset, and testing two algorithms did not add much extra time or work. It was also believed that this would give a more thorough evaluation of the features in case they worked better on one of the algorithms.

This was a good choice, because it turned out that some features (n-grams and topic models), produced better results with the GP algorithm, while others (GloVe and LIWC) delivered better with the SVR.

The best GloVe results were achieved by running GloVe on an SVR algorithm with 200 dimensions. However, this took up quite a lot of computation time. The experiment results showed that when the dimensions need to be smaller, for example when the amount of data is very large or the speed of the model is of importance, GP can be a good choice over SVR. However, in the case of this thesis, where the amount of data is fairly small and the time is not a big issue, the 200-dimensional GloVe model with SVR is the best choice for optimal performance.

9.1.3. Regression Models for Personality Detection

In the experiments where both SVR and GP were used, it became clear that the best performing algorithm depended on the feature at hand and the configurations chosen. It also depends on which feature it is most important to detect. SVR tended to perform better when the features were highly dimensional, while GP would often give better results when the input space was smaller. In the case of LIWC, where the GP trait results were good, the results using GP were far better than with SVR. However, when looking at all traits combined, SVR performed better. This means that even though GP had very good scores on the traits it was the best to predict (conscientiousness and emotional stability), it was not able to deliver good results for the other traits. The good results for conscientiousness and emotional stability were also produced with different GloVe dimensions (50 and 200 respectively). Based on the literature study of the related research, the most important personality traits for pro-ED classification were neuroticism (measured through emotional stability) and openness. SVR with GloVe was chosen because it produced the best overall results. It was also the algorithm that produced the best overall openness results. With emotional stability, the algorithm did not do that well, but it was not the worst out of the algorithms either. It was decided that a trade-off between a bit lower emotional stability score and a high openness score was acceptable. It can, however, be argued that ridge regression could also be a good choice when looking to find only emotional stability and openness. Ridge regression produced the best emotional stability score, but a somewhat lower openness score than SVR.

When comparing the results obtained from SVR, with GloVe as the feature, to the personality baseline (SVR with LIWC) all of the five personality traits improved greatly. The most notable improvement was for agreeableness, which went from having a score of -0.152 to 0.510. When comparing the results to the state-of-the-art personality detection model proposed by Arnoux et al. (2017), it can be seen that the extroversion score found in this thesis is slightly higher (from 0.25 to 0.288). Arnoux et al. (2017) use two decimals in their result representation while three are used in this thesis. The agreeableness score was improved by quite a lot (from 0.29 to 0.510). For the conscientiousness score, only a minor improvement was achieved (from 0.33 to 0.340). The emotional stability score

decreased from 0.42 to 0.166 and the openness score decreased from 0.37 to 0.284. This means that the personality detection model proposed in this thesis managed to improve three personality traits compared to a state-of-the-art detection model. As mentioned in chapter 5.2, Arnoux et al. (2017) used GP in combination with GloVe to achieve their results. This combination was also tested in this thesis and the results show that only agreeableness and conscientiousness got higher scores than what they did for Arnoux et al. (2017). This means that it was not possible to recreate the results Arnoux et al. (2017) got, even with the use of the same detection model and feature. One reason for this might be the dataset used in this thesis. First of all, the dataset used in this thesis was much smaller than that used by Arnoux et al. (2017). Second, the dataset consisted of a combination of two different datasets and two different types of text, while Arnoux et al. (2017) used one complete dataset consisting of the same type of text. If a larger and more complete dataset had been used in this thesis, it might have helped improve the emotional stability and openness score and given results that were easier to compare to those of Arnoux et al. (2017).

9.1.4. Pro-ED Dataset Label Distribution

The dataset used to train the pro-ED classification model was not a balanced dataset. Tweets labeled as unrelated made up 65.3% of the tweets in the dataset while pro-recovery made up 10.4% and pro-ED 24.3%. Even though more than half the dataset consists of tweets labeled as unrelated, the amount of pro-ED tweets is still too large to represent the actual distribution of pro-ED vs non-pro-ED accounts on Twitter. This means that the pro-ED dataset used to train and test the pro-ED classification model is not representative of actual Twitter data. If the pro-ED classifier created in this thesis was applied to actual Twitter data, the classifier could therefore be expected to produce at least some false positives due to it expecting a higher proportion of pro-ED accounts than what is actually the case on Twitter. Deciding how large the proportion of pro-ED accounts is on Twitter at a given time is almost impossible due to changing hashtag use and behaviors. It is, however, fair to assume that the proportion of pro-ED accounts is tiny compared to the total amount of Twitter accounts. This goes for pro-recovery accounts as well, especially when looking at pro-recovery accounts that only focus on eating disorder recovery.

9.1.5. Establishing a Pro-ED Result Baseline

When creating a pro-ED result baseline, a state-of-the-art pro-ED classification model proposed by Giæver (2018) was used as a starting point for the baseline. The starting point was created by recreating the results Giæver got with her state-of-the-art pro-ED classification model. The starting point was then improved upon by a re-annotation (described in chapter 6.2.2) as well as the eight pre-processing steps described in chapter 6.4. When comparing the starting point results to the results from the pre-processing steps, it becomes apparent that the pre-processing steps themselves were enough to improve the performance of the state-of-the-art pro-ED classification model. Compared to the results Giæver reported (as seen in table 8.12) there was an increase in precision

by 1.5%, in recall by 0.4% and in F_1 score by 0.9%. The starting point results (obtained from recreating Giævers pro-ED classification model with the same dataset) did, however, produce slightly better results than those reported by Giæver, meaning that the increase in measures is slightly lower when comparing the baseline (table 8.14) to the results from the recreation. When comparing the baseline to the recreated results there was no change in precision. Recall increased by 0.4% while F_1 score increased by 0.2%.

9.1.6. Features for Pro-ED Classification

When testing n-grams as a feature for pro-ED classification the measurement scores for bigrams (both precision, recall and F_1 score) gradually improved up to the use of 5000 features, after this the scores worsened. This is most likely because 300 and 900 features are too little and do not give enough information, while 5000 and more features might start to give too much information. This can then make the model overfit to the training data and therefore not perform as well on test data. Another thing worth mentioning from the n-gram experiments is that more features were needed to get the optimal scores for bigrams compared to unigrams. This could be because looking at two words in combination creates more features from the same text. From the result, it could also be observed that the results from bigrams were slightly worse than for unigrams, except for precision which was marginally better for some values of the maximum number of features. This suggests that most of the pro-ED accounts can be detected using single words. However, since precision got slightly better with the bigrams, it might be that bigrams detect when an account is using a term related to pro-ED in a different context more accurately than unigrams.

As for the other three features (topic models, POS tags and LIWC) only the topic models got good enough results to be included in the final classifier. When visualizing the topics in image 8.2 during the pre-processing experiments, it was clear that the topic models were able to detect many words related to eating disorders. Knowing this, it is not very surprising that this feature got good results. With POS on the other hand, it is reasonable to believe that the tags were very similar for the tweets. Considering the limited amount of characters, it is very likely that many of the tweets were similar in structure. With this in mind, it is impressive that POS was still able to get an F_1 score of 0.96. As for the LIWC experiments, it could be observed that the performance seemed to be best with between 15 and 80 features. This could mean that adding a high number of features creates unnecessary noise, while adding less than 15 gives too little information.

9.1.7. Choosing the Pro-ED Algorithm

In order to compare the different algorithms, it was decided to use unigrams as the only feature. The reason for this was that unigrams were proven to produce high performance when tested as a feature. Choosing the same feature for all the algorithms also meant that it would be easier to compare the results. The experiments that looked at testing the different machine learning algorithms for pro-ED classification produced very good results

for all the algorithms. The best results were found with the Support Vector Machine (SVM) and the Multilayer Perceptron (MLP) algorithms, which both achieved F_1 scores above 0.99. Seeing these two algorithms perform well makes sense, as the input was highly dimensional and the algorithms are known to be good at handling this type of input. The worst performance was achieved by Gaussian Process (GP). This could also be caused by the high-dimensional input, which is not optimal for GP. This was also supported by the personality detection GloVe experiments, where GP got better results at lower data dimensions. The results for K-Nearest Neighbors (k-NN) and Ridge Regression (RR) were both good, but not as good as the SVM or the MLP. This could be because these models are too simple for this type of problem, or because there is too much noise in the data for them to generalize well. It can be observed that k needed to be quite high for the k-NN algorithm in order for it to reach its optimal values, which could support the case that there was a bit too much noise in the data.

All the results produced by the Multilayer Perceptron (MLP) algorithm were at, or above, 0.99 in F_1 score. Because these results seemed almost too good to be true, the results were manually tested. It turned out that when the layer size was set to 50, only 4 accounts were mis-classified and that these were all unrelated accounts classified as pro-ED. This means that the recall was indeed 100% and precision 99%. Seeing how good these results were, and considering that the algorithm was fast to both train and predict, MLP was chosen as one of the algorithms to test for the final classifier. The algorithm experiments were not cross-validated, as it was with the feature experiments, but rather trained and validated multiple times using random splits of the training set for training and validation. This might be part of the reason for the unusually high performance scores on this experiment. Yet, when considering this was done for all the algorithms, it still proves the MLP algorithm to be superior.

9.1.8. Pro-ED with Personality as Part of a Feature Set

All in all, the final classifier experiment shows that using personality as one of the features in a feature set will indeed increase the performance of the pro-ED classifier. In particular, the precision increased for both MLP and SVM. Adding personality as a feature either improved the performance of the classifier or left it as accurate as it was without it. The results for the final pro-ED classifier with MLP (table 8.24) delivered the best results. The results were, however, poorer than the results obtained when testing the algorithm with only unigrams as a feature (table 8.23) as part of the algorithm selection process. The reason for this can be that the final pro-ED classifier was tested on the test set that was created right after the dataset pre-processing. This meant that the algorithm was tested on completely new and unseen data. When testing the algorithm with only unigrams, the algorithm was trained on the training set and a small part of the training set was used as a test set. The same results were observed when testing the final classifier with only unigrams. Here the results (seen in table A.10 in appendix A) were poorer than those obtained from testing the algorithm with the training set.

When comparing the results obtained by using MLP with the new feature set (table 8.24) to the baseline (table 8.14), it can be seen that the precision decreased by 0.2%, the recall increased by 0.2% and the F_1 score increased by 0.1%. This means that there was a slight improvement from the baseline to the new state-of-the-art pro-ED classification model.

9.1.9. Pro-ED Multi-Class Results

A multi-class version of the binary pro-ED classification baseline was created as a second baseline. When comparing the binary baseline (table 8.14) to the multi-class baseline (table 8.15) it could be seen that the scores for the multi-class baseline were lower than the binary baseline, which is no surprise. With the multi-class, baseline there were three labels to predict, instead of two, and the number of tweets belonging to each label was highly uneven. The number of tweets belonging to the pro-recovery label amounted to only 10.4% of the total dataset while, in comparison, 65.3% of the tweets were labeled as unrelated and 24.3% as pro-ED. One thing that is worth taking note of is that one feature, the Twitter account bio, produced better results in this experiment than it did with the binary baseline. This could be because much of the information on pro-recovery Twitter accounts are given in the bio, and hence, this feature could be better at identifying the pro-recovery class.

The final version of the binary pro-ED classifier was also run as a multi-class classifier, both with and without personality. The results from the final multi-class classifier show that personality did, in fact, increase the performance of the classifier compared to the results obtained when personality was excluded (this finding was apparent for the binary classifier as well). On the other hand, the results also showed that the feature set used in the final pro-ED classifier was not as good as the feature set used for the baseline creation when running the classifier as a multi-class. When considering that the bio alone gave better results for the multi-class classification than it did in the binary classification, this feature could have been important in the baseline classifier. This might be what caused the overall decrease in the performance of the final multi-class classifier. The best multi-class results were achieved by running MLP with personality as a feature. These results were identical to the multi-class baseline results (table 8.15), so even though the overall performance decreased with the new feature set, at least the best results could compete with the baseline results.

9.2. Ethics

The data handled during the course of this thesis consist of real human thoughts and opinions. While some of the datasets consist of data willingly provided by the owners, some datasets, such as the pro-ED dataset, consist of data accumulated without the knowledge or permission of the owner. As a result, it became important to consider the ethical aspects of working with such data. The datasets were handled only by the two

authors, making the exposure of data as small as possible. Care was taken to make sure that no information about Twitter account owners in the thesis would create a possibility of revealing the owner. When tweet examples were used to illustrate a point, it was made sure that it would not be possible to identify the tweet owner from the tweet. This was done by making changes to the tweets, such as censoring any mention of account usernames or what appeared to be real names, as well as make small structural changes while still containing the meaning and value of the tweet. In rare cases where real data could not be used due to ethical or copyright reasons, fictive examples were crafted in order to illustrate the point. The Twitter profile and tweet examples from chapter 2.3 are such fictive examples, where creating a fake Twitter account helped bypassing the problems of displaying a real pro-ED account.

Another major ethical problem revolved around the effect that viewing pro-ED content might have on the authors. As mentioned in chapter 5.5, a study done by Csipke and Horne (2007) found that viewing pro-ED content had a high possibility of affecting the viewer in a negative way. With this in mind, it was decided, at the beginning of working with this thesis, that the authors would keep a close eye on each other and communicate regularly about challenges, elements that seemed difficult or things that might have affected the authors in one way or another. This was especially important during the annotation process, where thousands of pro-ED accounts were evaluated. One of the authors reported to be shocked by the content severity of some of the accounts and that it was emotionally draining to do the annotation. None of the authors has, however, experienced any long term effects of viewing pro-ED content.

9.3. Limitations

A possible limitation to the potential of the pro-ED dataset was that the pro-ED dataset had already been modified to some extent by Giæver (2018). These modifications were done in a slightly different manner than what might have been optimal for the experiments in this thesis. It had also introduced some inconsistency to the dataset with encoded byte strings and number deletion (as described in chapter 6.4). The byte strings were decoded whenever possible. The byte strings that were impossible to decode, due to number removal in tweet text, were deleted completely. This deletion might have resulted in a small amount of data loss when character combinations such as *XD* or *x* were deleted. These symbols are often used as an emoji with crossed eyes and a huge smile or used to mean *kiss*, respectively. The symbols mentioned are, however, rarely used compared to the amount of noise that was removed from the dataset, which is why it was decided that this was OK. Nevertheless, this does pose a chance of minor errors in the dataset.

The Big 5 personality detection model was trained on a separate dataset to the pro-ED dataset. Ideally, the personality model should have been trained on the pro-ED dataset, but unfortunately, this was not possible due to the pro-ED dataset not containing Big 5 personality trait labels. By using a different dataset to train the personality detection

model it is difficult to determine how accurate the personality result for the pro-ED dataset is. The only way the personality model was measured was through the pro-ED classification model and whether the personality feature improved the results. Being able to see exactly how the personality model performed on the pro-ED dataset would have helped understand how the personality feature affected the pro-ED classification model.

Two different datasets were used to train the personality detection model. Three were initially intended to be used, but the YouTube transcriptions dataset was removed, as explained in chapter 9.1.1. The two remaining datasets came from the same source and had therefore corresponding formats. They also used the same method for creating the Big 5 personality trait scores. Even though all the datasets were labeled with the Big 5 personality model traits, differences in personality estimation might occur and can, therefore, lead to errors in the experimental results. Measures were, however, taken in order to try to measure and minimize these errors, as explained in chapter 6.3.

The five machine learning algorithms tested for pro-ED classification (described in chapter 8.2.4) were tested without the use of cross-validation. The reason for this was that the built-in cross-validation method in Scikit-learn was unable to handle non-linear models, and due to time restrictions, it was not possible to implement a method that could. This means that the algorithms were run on the same training and validation split for each test, even if each experiment was done several times, which could lead to results being non-representative because of chance.

10. Conclusion and Future Work

The thesis conclusion is presented in this chapter. An evaluation of the thesis goal and how it was met is first described. This is done by presenting the findings for each of the three research questions. Pro-ED classification of Twitter accounts is a narrow research field and the contributions of this thesis are presented next. The last part of this chapter contains ideas for what could be done in the future to continue or improve upon the work presented in this thesis.

10.1. Goals and Research Questions

The goal for this thesis was:

Goal *To improve upon automatic detection of pro-ED Twitter accounts by considering personality as a feature.*

This goal was re-shaped into three research questions, which had the function of guiding the research in the direction of the goal. The answer to each research question would lead the research into the phase of the next research question. The last research question, however, was constructed to deliver a measurement of whether the research goal had been achieved. The first research question was formulated as follows:

Research question 1 (Personality) *Which machine learning model has the best potential for personality detection?*

Through experiments focusing on feature extraction and performance comparison of five different machine learning algorithms it was found that the Support Vector Regression algorithm delivered the best result when executed with Global Vectors as the feature.

Research question 2 (Pro-ED) *Which machine learning model has the best potential for pro-eating disorder classification?*

Research question 2 was similar to the first research question, with the only difference being the focus on pro-ED Twitter account classification instead of personality detection. As with the first research question, feature extraction experiments were carried out in

order to find the features most suitable for pro-ED classification. These features were tested on five machine learning algorithms. The machine learning algorithm that delivered the best results was the Multilayer Perceptron with unigrams as the feature and a layer size of either 50 or 200.

Research question 3 *What impact does the inclusion of personality detection, as a feature, have on the performance of the pro-ED classifier?*

The third and final research question was designed to take the results from the two previous research questions and combine them, through experiments, into one complete pro-ED classification model. The predictions from the personality detection model were used as a feature for the pro-ED classification model. The Multilayer Perceptron pro-ED classification model delivered the best results when personality was included in the feature set compared to running the algorithm without personality. The difference was an increase in F_1 score by 0.3%. The feature set consisted of unigrams, bigrams, and topic models.

10.2. Contributions

The experiments focusing on personality detection contribute to the expansion of the personality detection research field by using state-of-the-art personality detection methods in a new way. A new dataset consisting of different types of data has been created. New and different features from what have been used in state-of-the-art personality detection models have been tested. Some models were found to be better at detecting parts of the Big 5 personality traits than others, which contributes to the search for a model that has a high performance on a subset of the Big 5 personality traits.

The Pro-ED oriented work presented in this thesis contributes to the limited amount of research that exists on the detection of pro-eating disorder in Twitter accounts, in the detection of eating disorders online in general and to the text classification research field. A new and more consistent pro-ED dataset has been created with annotations thoroughly checked for errors and subjective choices. The process of categorizing accounts, as either pro-ED, pro-recovery and unrelated, presented by Giæver (2018) has been verified and deemed successful. Furthermore, the work presented by Giæver has been tested and strengthened by recreating her experiments with success. The work presented in this thesis shows that classification of pro-ED Twitter accounts can be done with an F_1 score of 0.99, which is an improvement on the only state-of-the-art model in existence. Hopefully, this can contribute to reaching out to people in need of help in a quick and effective way. The detection of pro-ED accounts is also an important stepping stone into removing content online that might cause harm to people.

10.3. Future Work

As mentioned in chapter 9.1.4 the dataset used for pro-ED classification was an unbalanced dataset. It would be interesting to grow the dataset to contain a larger portion of unrelated accounts. This is to give a more accurate representation of the account distribution on Twitter. The pro-ED dataset used in this thesis contains a much larger portion of pro-ED accounts than what is expected to be in the total amount of accounts on Twitter. This should be tackled as it is a possible error source, especially when considering to use the classification model on live data from Twitter. Growing the dataset to contain more pro-recovery labeled accounts could also help make the pro-ED multi-class classification model more accurate than what was found in this thesis. Another addition to the dataset would be adding more accounts that are in the *grey area* between pro-ED and unrelated, considering the accounts in this dataset are mostly on the ends of the spectrum.

Another thing that would be interesting is to use a dataset that is annotated for both personality and pro-ED, and run the experiments in this thesis on the dataset. This might yield more accurate and measurable results, at least in regards to the personality detection. This means that the dataset would have to be created with the consent of the pro-ED Twitter account owners (both to use their accounts and to measure their personality), which minimizes the ethical aspects of this type of research.

Looking at image recognition as a feature for pro-ED classification could possibly improve the classification model. Many pro-ED users post thinspo pictures, something the classification model presented in this thesis does not take into account. If it was possible to detect these thinspiration images, it could be a very promising feature to use in the classification model. Another feature that could be interesting to look at is the text in URLs. It could be that many URLs posted in pro-ED account tweets contain ED-related words. In the pro-ED dataset used in this thesis, URLs were already replaced by a placeholder, making this impossible. Other metadata, such as Twitter account followers, followed accounts and account relationships, could provide insight into how the pro-ED community is constructed, interacts and reaches out to new members.

Using hashtags to see if it is possible to detect pro-ED Twitter accounts would also be an interesting research topic. By looking at the evolution of the hashtags used in pro-ED Twitter accounts, and the correlation between hashtags, it might be possible to detect a pattern in the evolution of the hashtags and the connection between them. If this is done successfully then it might be possible to detect pro-ED accounts hiding behind hashtags with a secret meaning. It might also be possible to detect new trends in the pro-ED community quickly. The evolution of a Twitter account user over time is also an interesting research topic. During the annotation process there were a number of times where the authors were unsure of whether the account owner had moved from being pro-ED to pro-recovery or vice versa. By looking at the evolution of tweet content it might be possible to get a clearer distinction between pro-ED and pro-recovery.

Bibliography

Pierre-Hadrien Arnoux, Anbang Xu, Neil Boyette, Jalal Mahmud, Rama Akkiraju, and Vibha Sinha. 25 tweets to know you: A new model to predict personality with social media. *arXiv preprint arXiv:1704.05513*, 2017.

Alina Arseniev-Koehler, Hedwig Lee, Tyler McCormick, and Megan A Moreno. #Proana: Pro-eating disorder socialization on Twitter. *Journal of Adolescent Health*, 58(6): 659–664, 2016.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

Mitja D Back, Juliane M Stopfer, Simine Vazire, Sam Gaddis, Stefan C Schmukle, Boris Egloff, and Samuel D Gosling. Facebook profiles reflect actual personality, not self-idealization. *Psychological science*, 21(3):372–374, 2010.

Anna M Bardone-Cone and Kamilla M Cass. What does viewing a pro-anorexia website do? An experimental examination of website exposure and moderating effects. *International Journal of Eating Disorders*, 40(6):537–548, 2007.

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. " O'Reilly Media, Inc.", 2009.

Laird Birmingham, Jenny Su, Julia A Hlynsky, Elliot M Goldner, and Min Gao. The mortality rate from anorexia nervosa. *International Journal of Eating Disorders*, 38 (2):143–146, 2005.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Natalie Boero and Cheri Jo Pascoe. Pro-anorexia communities and online interaction: Bringing the pro-ana body online. *Body & Society*, 18(2):27–57, 2012.

Elke Bollen and Franz L Wojciechowski. Anorexia nervosa subtypes and the big five personality factors. *European Eating Disorders Rewiew*, 12(2):117–121, 2004.

Dina LG Borzekowski, Summer Schenk, Jenny L Wilson, and Rebecka Peebles. e-ana and e-mia: A content analysis of pro–eating disorder web sites. *American Journal of Public Health*, 100(8):1526–1534, 2010.

John D Burger, John Henderson, George Kim, and Guido Zarrella. Discriminating gender on Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1301–1309. Association for Computational Linguistics, 2011.

Antonio A Casilli. Online networks of eating-disorder websites: Why censoring pro-ana might be a bad idea. *Perspectives in Public Health*, 133(2):94–95, 2013.

Fabio Celli, Fabio Pianesi, David Stillwell, and Michal Kosinski. Workshop on computational personality recognition (shared task). In *Proceedings of the Workshop on Computational Personality Recognition*, 2013.

Salvador Cervera, Francisca Lahortiga, Miguel A Martínez-González, Pilar Gual, Jokin de Irala-Estévez, and Yolanda Alonso. Neuroticism and low self-esteem as risk factors for incident eating disorders in a prospective cohort study. *International Journal of Eating Disorders*, 33(3):271–280, 2003.

Stevie Chancellor, Yannis Kalantidis, Jessica A Pater, Munmun De Choudhury, and David A Shamma. Multimodal classification of moderated online pro-eating disorder content. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3213–3226. ACM, 2017.

Laurence Claes, Walter Vandereycken, Patrick Luyten, Bart Soenens, Guido Pieters, and Hans Vertommen. Personality prototypes in eating disorders based on the big five model. *Journal of Personality Disorders*, 20(4):401–416, 2006.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Glen Coppersmith, Mark Dredze, Craig Harman, Kristy Hollingshead, and Margaret Mitchell. CLPsych 2015 shared task: Depression and PTSD on Twitter. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 31–39, 2015.

Paul T Costa and Robert R McCrae. The revised neo personality inventory (neo-pi-r). *The SAGE Handbook of Personality Theory and Assessment*, 2(2):179–198, 2008.

Emese Csipke and Outi Horne. Pro-eating disorder websites: Users' opinions. *European Eating Disorders Review: The Professional Journal of the Eating Disorders Association*, 15(3):196–206, 2007.

Joseph L Fleiss, Bruce Levin, and Myunghee C Paik. *Statistical Methods for Rates and Proportions*. John Wiley & Sons, 2013.

Nick Fox, Katie Ward, and Alan O'Rourke. Pro-anorexia, weight-loss drugs and the internet: An 'anti-recovery'explanatory model of anorexia. *Sociology of Health & Illness*, 27(7):944–971, 2005.

Jeff Gavin, Karen Rodham, and Helen Poyer. The presentation of "pro-anorexia" in online group interactions. *Qualitative Health Research*, 18(3):325–333, 2008.

Ata Ghaderi and Berit Scott. The big five and eating disorders: A prospective study in the general population. *European Journal of Personality*, 14(4):311–323, 2000.

Ingrid N Giæver. Classification of pro-eating disorder users on Twitter. MSc Thesis, Dept. of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway, June 2018.

Jennifer Golbeck, Cristina Robles, Michon Edmondson, and Karen Turner. Predicting personality from Twitter. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 149–156. IEEE, 2011a.

Jennifer Golbeck, Cristina Robles, and Karen Turner. Predicting personality with social media. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 253–262. ACM, 2011b.

Adrienne S Juarascio, Amber Shoaib, and Alix Timko. Pro-eating disorder communities on social networking sites: A content analysis. *Eating Disorders*, 18(5):393–407, 2010.

Tom Kenter and Maarten De Rijke. Short text similarity with word embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1411–1420. ACM, 2015.

Upendra Kumar, Aishwarya N Reganti, Tushar Maheshwari, Tanmoy Chakroborty, Björn Gambäck, and Amitava Das. Inducing personalities and values from language use in social network communities. *Information Systems Frontiers*, pages 1–22, 2017.

Chenglong Ma, Weiqun Xu, Peijia Li, and Yonghong Yan. Distributional representations of words for short text classification. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 33–38, 2015.

Chris Mann and Fiona Stewart. *Internet Communication and Qualitative Research: A Handbook for Researching Online*. Sage, 2000.

Robert R McCrae and Oliver P John. An introduction to the five-factor model and its applications. *Journal of Personality*, 60(2):175–215, June 1992.

Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. Efficient estimation of word representations in vector space. pages 1–12, 01 2013.

Margaret Mitchell, Glen Coppersmith, and Kristy Hollingshead, editors. *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, Denver, Colorado, June 5 2015. Association for Computational Linguistics. doi: 10.3115/v1/W15-12. URL `https://www.aclweb.org/anthology/W15-1200`.

Alexandre Passant, Tuukka Hastrup, Uldis Bojars, and John Breslin. Microblogging: A semantic web and distributed approach. 2008.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of LIWC2015. Technical report, 2015.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

Pinterest. User guidelines, 2012. URL `https://policy.pinterest.com/nb/community-guidelines`. Accessed 2018-10-23.

John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, April 1998. URL `https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/`.

Daniel Preoţiuc-Pietro, Maarten Sap, H Andrew Schwartz, and Lyle Ungar. Mental illness detection at the world well-being project for the CLPsych 2015 shared task. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 40–45. Association for Computational Linguistics, 2015.

Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

Philip Resnik, William Armstrong, Leonardo Claudino, Thang Nguyen, Viet-An Nguyen, and Jordan Boyd-Graber. Beyond LDA: Exploring supervised topic modeling for depression-related language in Twitter. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 99–107, 2015.

Shalom H Schwartz. Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In *Advances in Experimental Social Psychology*, volume 25, pages 1–65. Elsevier, 1992.

R Sudhesh Solomon, PYKL Srinivas, Amitava Das, Bjorn Gamback, and Tanmoy Chakraborty. Understanding the psycho-sociological facets of homophily in social network communities. *IEEE Computational Intelligence Magazine*, 14(2):28–40, 2019.

Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The general inquirer: A computer approach to content analysis. 1966.

Jorunn Sundgot-Borgen and Monica K Torstveit. Prevalence of eating disorders in elite athletes is higher than in the general population. *Clinical Journal of Sport Medicine*, 14(1):25–32, 2004.

Serra S Tekiroglu, Gözde Özbal, and Carlo Strapparava. Sensicon: An automatically constructed sensorial lexicon. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1511–1521, 2014.

Tumblr. A new policy against self-harm blogs, 2012. URL `https://staff.tumblr.com/post/18132624829/self-harm-blogs`. Accessed 2018-10-23.

Ernest C Tupes and Raymond E Christal. Recurrent personality factors based on trait ratings. Technical report, Personnel Research Lab Lackland AFB TX, 1961.

Twitter. Q2 2018 letter to shareholders, 2018a. URL `https://investor.twitterinc.com/static-files/610f4a82-5b52-4ed9-841c-beecbfa36186`. Accessed 2018-10-23.

Twitter. How to tweet, 2018b. URL `https://help.twitter.com/en/using-twitter/how-to-tweet`. Accessed 2018-10-26.

Krista Whitehead. Hunger hurts but starving works: A case study of gendered practices in the online pro-eating-disorder community. *Canadian Journal of Sociology*, 35(4): 595–626, 2010.

Jenny L Wilson, Rebecka Peebles, Kristina K Hardy, and Iris F Litt. Surfing for thinness: A pilot study of pro–eating disorder web site usage in adolescents with eating disorders. *Pediatrics*, 118(6):e1635–e1643, 2006.

Michael Wilson. MRC psycholinguistic database: Machine-usable dictionary, version 2.00. *Behavior Research Methods, Instruments, & Computers*, 20(1):6–10, 1988.

Daphna Yeshua-Katz and Nicole Martins. Communicating stigma: The pro-ana paradox. *Health Communication*, 28(5):499–508, 2013.

A. Experiment Results

This appendix contains tables showing the results of experiments that were not added to the chapter 8. Each of the tables present in this appendix is described individually.

Table A.1.: GloVe Results when Including the YouTube Dataset

GloVe dimension	**Algorithm**	**Extroversion**	**Agreeableness**	**Conscientiousness**	**Emotional stability**	**Openness**
200	GP	0.268	**-0.070**	-0.226	-0.237	0.281
200	SVR	**0.297**	-0.079	**-0.172**	**-0.147**	**0.298**

Table A.1 presents the results that were achieved when including the YouTube dataset in calculating the personality. The experiment used GloVe as the feature with 200 dimensions, and results are given using both Gaussian Process (GP) and Support Vector Regression (SVR). These results can be compared to table 8.3, where the results of running GloVe without the YouTube dataset is displayed.

Table A.2.: Personality Results of Different N-Grams

N	**Algorithm**	**Extroversion**	**Agreeableness**	**Conscientiousness**	**Emotional stability**	**Openness**
1	GP	**0.109**	-0.045	0.030	0.125	0.124
	SVR	0.078	0.006	0.004	0.151	0.039
2	GP	0.017	0.085	0.008	0.052	0.153
	SVR	0.009	0.048	0.009	0.017	0.132
char 5-10	GP	0.099	0.079	**0.084**	**0.277**	**0.193**
	SVR	0.041	**0.103**	0.081	0.196	0.123

Table A.2 contains the results of running the SVR and the GP algorithms on the personality dataset with different n-grams as feature.

Table A.3.: Personality LIWC Results with SVR

N	**Extro-version**	**Agreea-bleness**	**Conscien-tiousness**	**Emotional stability**	**Openness**
5	0.153	-0.179	-0.008	0.099	0.045
10	0.163	0.033	-0.008	0.152	0.054
15	0.143	-0.039	-0.010	0.227	0.048
20	0.141	0.112	-0.009	0.174	0.048
25	0.173	-0.019	0.007	0.201	0.051
30	0.109	-0.182	0.002	0.220	0.058
35	**0.189**	0.093	-0.007	0.198	0.037
40	0.077	0.080	0.005	0.105	0.054
50	0.032	-0.203	**0.078**	-0.287	0.087
60	0.097	**0.255**	0.075	**0.332**	0.134
70	0.050	0.243	0.020	0.318	0.110
80	-0.019	-0.175	0.078	-0.03	0.146
94	0.153	-0.152	0.024	0.034	**0.168**

Table A.3 shows the complete results of the experiments using Linguistic Inquiry and Word Count (LIWC) as a feature for personality. The algorithm used in this experiment was SVR. This table is meant as an extension to the table 8.7 presented in chapter 8.1.3.

Table A.4.: Pro-ED Unigrams Feature Results with Various Number of Features

Features	Precision	Recall	F1
300	0.994	0.982	0.988
900	**0.995**	0.984	**0.990**
1200	**0.995**	**0.986**	**0.990**
2000	**0.995**	0.985	**0.990**
5000	**0.995**	0.985	**0.990**
10000	**0.995**	0.984	**0.990**
15000	**0.995**	0.984	**0.990**
20000	**0.995**	0.984	**0.990**

Table A.4 displays the complete results of all tests using different numbers of features for unigrams with the pro-ED classifier, as described in chapter 8.2.3.

Table A.5.: Pro-ED Bigrams Feature Results with Various Number of Features

Features	Precision	Recall	F1
300	0.988	0.955	0.971
600	0.990	0.963	0.976
900	0.989	0.966	0.977
1200	0.994	0.965	0.979
1500	0.994	0.968	0.981
5000	**0.996**	**0.971**	**0.984**
10000	**0.996**	0.968	0.981
15000	0.994	0.967	0.980

Table A.5 presents the complete results of all tests using different numbers of features for bigrams with the pro-ED classifier, as described in chapter 8.2.3.

Table A.6.: Pro-ED Topic Model Feature Results with Different Number of Topics

Method	Num topics	Precision	Recall	F1
BoW	5	0.954	0.899	0.925
BoW	10	**0.989**	**0.990**	**0.989**
BoW	15	0.987	**0.990**	0.988
BoW	20	0.974	0.989	0.982
TF-IDF	5	0.877	0.980	0.925
TF-IDF	10	0.972	0.989	0.980
TF-IDF	15	0.967	0.984	0.975
TF-IDF	20	0.935	0.982	0.958

Table A.6 shows the complete results of all tests using different numbers of topics for the topic model experiments with the pro-ED classifier. It shows the results from using both the Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) models as input for all the numbers of topics that were tested.

Table A.7.: Pro-ED POS Feature Results

n	Precision	Recall	F_1
1	0.899	0.896	0.897
2	0.947	0.934	0.941
3	0.962	**0.949**	0.956
4	0.970	0.948	0.959
5	**0.972**	0.948	**0.960**
6	0.965	0.940	0.952
7	0.954	0.921	0.937

Table A.7 contains the results of testing POS as feature with the SVM algorithm. The POS tags were considered as an n-gram model on character level. Different values for n were tested and measured through precision, recall and F_1 score, as described in chapter 8.2.3.

Table A.8.: Pro-ED SVM Results with Different C-Values

C	Precision	Recall	F1
0.01	0.985	0.915	0.949
0.1	0.994	0.969	0.981
0.5	0.995	0.982	0.989
1	0.995	0.984	0.990
1.5	**0.996**	0.985	0.990
2	**0.996**	0.985	**0.991**
3	0.995	0.986	**0.991**
5	0.995	0.986	0.990
7	0.995	0.987	**0.991**
10	0.995	**0.988**	**0.991**

Table A.8 shows the results of experimenting with different values for C in the Support Vector Machine (SVM) algorithm for the pro-ED classifier. The feature used was unigrams.

Table A.9.: Pro-ED GP Results with Different Kernels

Kernel	Precision	Recall	F_1
DotProduct+ WhiteKernel	**0.987**	**0.956**	**0.971**
RBF	**0.987**	0.948	0.967

Table A.9 contains the results of running the GP algorithm with different kernels. Two different kernels were tested, namely the dot-product kernel and the white-kernel.

Table A.10.: Results for Final Classifier with Single Features

Algorithm	Features	Precision	Recall	F1
MLP	unigrams	0.995	0.982	0.988
SVM	unigrams	0.995	0.979	0.987
MLP	bigrams	0.993	0.975	0.984
SVM	bigrams	0.993	0.959	0.976
MLP	topic models	0.991	0.973	0.982
SVM	topic models	0.986	0.973	0.979
MLP	personality	0.761	0.604	0.673
SVM	personality	0.727	0.616	0.667

Table A.10 presents the results of running the final pro-ED classifier on the test set with only one feature at a time. The table presents the results from both the Multilayer Perceptron (MLP) algorithm and the SVM algorithm with each of the individual features.

Andrea Hollung Nornes, Martine Alvilde Gran

Automatic Classification of Pro-Eating Disorder Twitter Accounts with Personality as a Feature

NTNU
Kunnskap for en bedre verden